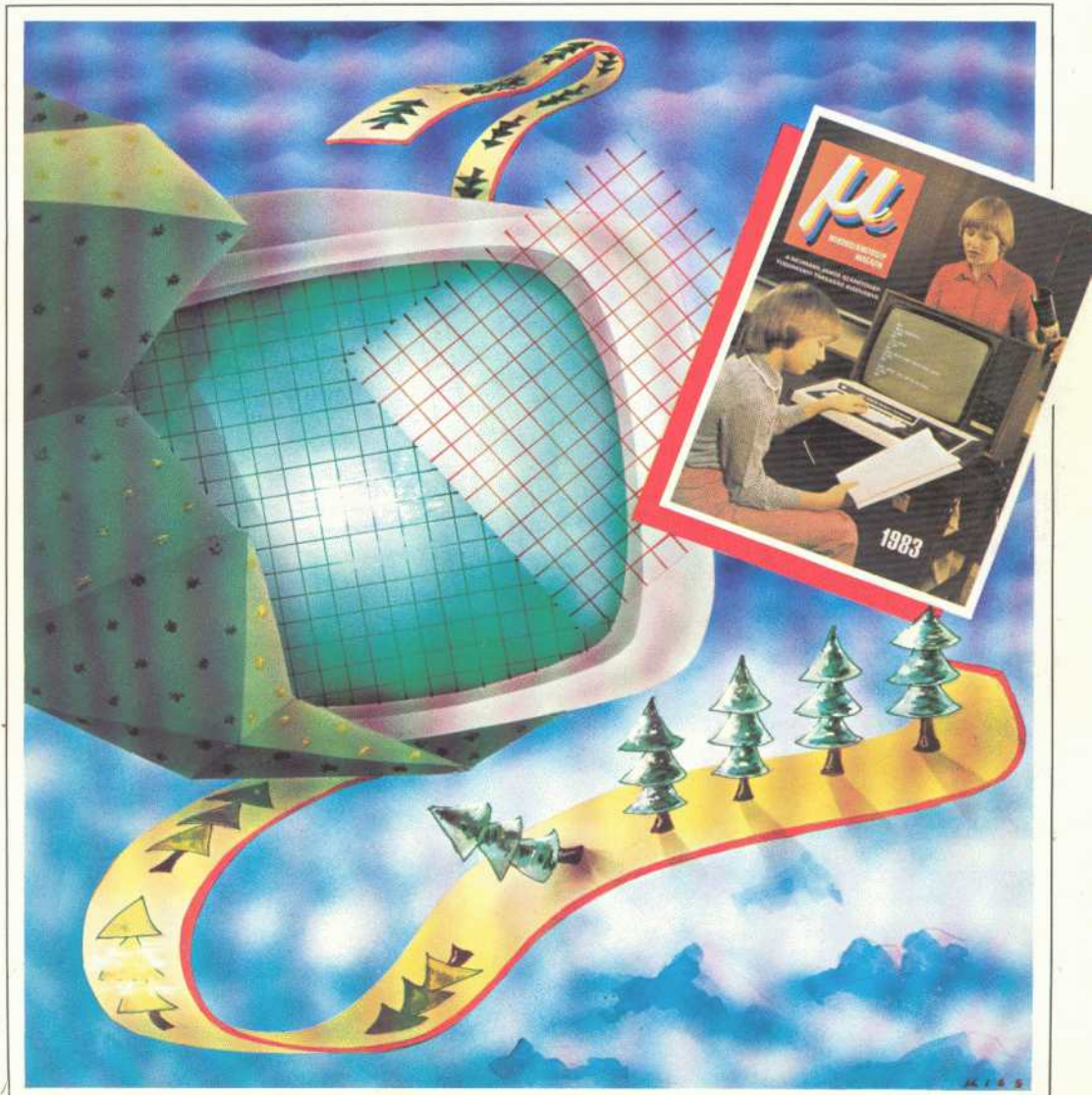
Ára: 30 Ft

mikro számítógép magazin



LETTÜNK, ÍRTUNK! GYŐZTÜNK?

1988/12

NINCSEN BASIC

TÖVIS NÉLKÜL . . .

μ mikro számítógép magazin

6. ÉVFOLYAM
1988/12. SZÁM

A NEUMANN JÁNOS SZÁMÍTÓGÉPTUDOMÁNYI TÁRSASÁG LAPJA

A szerkesztőbizottság vezetője:
Kovács Győző

A szerkesztőség munkatársai:
Bakos Tamás (programozástechnika)
Broczkó Péter (hírek)
Ferencz Mária (tervezőszerkesztő)
Kovács Győző (levelezés)
Petróczy Judit (könyvek)
Pinke György (NJSZT, alkalmazások)
Simonyi Endre
Szebenszki Sándor
Szulyovszky Csaba
Tamásné Lakó Erika
Terebessy Ákosné
Vizesi Mária

Címképünk:
Kiss Ilona munkája

Felelős szerkesztő:
Könyves Tóth Pál

Szerkesztőség:
1027 Budapest, Fő u. 68.
Telefon: 154-250

Levélcím:
1371 Budapest
Pf. 433.

Kiadja az Ifjúsági Lap- és Könyvkiadó Vállalat

Felelős kiadó:
dr. Király G. István
igazgató

Kiadóhivatal:
1065 Budapest, Révay u. 16.
Telefon: 116-660

Terjeszti a Magyar Posta
Előfizethető a hírlapkézbesítő hivataloknál
és a Posta Hírlap-előfizetési és Lapellátási Irodáján
(1900 Budapest XIII., Lehel u. 10/A)
vagy átutalással a 215—96 162 pénzforgalmi jelzőszámra.

Megjelenik havonta.
Egy szám ára 30,— Ft
Előfizetési díj:
egy évre 360,— Ft
fél évre 180,— Ft
Külföldön terjeszti a Kultúra,
1389 Budapest, Pf. 149.
és a Magyar Média
1932 Budapest, Pf. 279.
88—1552

Szikra Lapnyomda
Budapest (88—1669)
Felelős vezető:
Csöndes Zoltán vezérigazgató

INDEX: 25 629
ISSN 0236-6088

TARTALOM

2 Ötéves a Mikroszámítógép Magazin
6 Feladatok — megoldások
11 Egy sarokkal olcsóbb!
18 Monitorológia
20 Mókuskerék
21 SEIKOSHA SP—1200 VC
28 Rendszerfejlesztési eszközök
31 Tranzisztorországban
34 Merre tart a világ?
39 μINFORM
40 Olvastunk
45 Programtermék
46 Adok—veszek—cserélek

ISKOLA-SZÁMÍTÓGÉP 3
3 Statisztikai próbák programozása

CSIPEGETŐ 7
7 Digitális montázskészítő
7 RESET sajátkezűleg
8 Korszakváltás, de hogyan . . .?!
8 Magyar nyelvű hibaüzenetek
9 Mi miből származik, avagy a nyelvcsaládfáról . . .
10 Mandelbrot

PROGRAMOZÁSTECHNIKA 12
12 BASIC és gépi kód
13 Hibajelzések és üzenetek
15 A nagyfelbontású képernyő forgatása bájtonként
17 Környezetvédő „ZÖLD" rutinok . . .

ENTERPRISE 24
24 POKE-kal vagy POKE nélkül?
25 Megkérdeztük az ENTERPRISE-ról
26 Az ellipszis és a többiek
27 Mi a manó?

μPROGRAMOK 36
36 Segíts magadon!

μKLUB 38
38 Klubszervezés és oktatás
38 Adom a magyarázatot!

SAKK 42
42 Anyag és mozgékonyság

AZ OLVASÓ ÍRJA 43
HÍREK—ÉRDEKESSÉGEK, KÖNYVEK 46
PONTVADÁSZAT 48

ÖTÉVES A MIKROSZÁMÍTÓGÉP MAGAZIN

Az öregedésnek is — mint minden másnak — megvan a maga kellemes és kellemetlen oldala. Az utóbbit az ember igyekszik elfelejteni, míg az előbbit élvezi. Az előbbihez tartozik például, hogy mostanában meglehetősen sok helyre hívnak meg, csak előadást tartani, de évfordulókat ünnepelni is. Megvallom, minden ilyen alkalomnak nagyon örülök. Így nagy örömet szerzett a CHIP magazin levele is, amelyben meghívtak, hogy vegyek részt a lap alapításának tizedik évfordulója alkalmából rendezett ünnepi összejövetelen.

A CHIP a legnagyobb példányszámú (vagy az egyik legnagyobb példányszámú) európai számítógép-magazin, az első — egy összevont kettős — száma 1978 szeptemberében jelent meg, éppen öt évvel előzve meg a Mikroszámítógép Magazin kiadását.

Az ünnepség méltó volt egy százhúszezres sikerlaphoz. A vendégek lehettek vagy ezren. A hangulat kitűnő volt, talán csak az ünnepi szónoklatokkal bántak csinján a rendezők.

Dr. Kurt Eckernkamp úr, a Vogel Verlag egyik tulajdonosa bensőséges szavakkal méltatta a lap jelentőségét, a német számítástechnikában vállalt szerepét. Elmondta, hogy mit is jelentett és jelent még ma is egy ilyen példányszámú, elsősorban az „amatőröknek" és kevésbé a szakmának szóló újság. Persze annak ellenére, hogy a lap a nagyközönségnek készül — saját tapasztalatom szerint is — a szakma nagyon is szereti és elfogadja a lapot, ami valószínűleg a lap tárgyilagos írásainak, az elfogulatlan hardver-szoftver teszteknek, az újdonságokról szóló írásoknak és nem utolsósorban annak a ténynek köszönhető, hogy a CHIP nem a cégeknek, hanem a szakmának az elkötelezettje.

Az évforduló ünneplésére Münchenben megjelentek — nyugodtan mondhatom — a szakma legjelesebb európai képviselői — magamat persze nem számítom ide, hiszen minden rendszerben vannak kivételek is —, a legfontosabb számítástechnikai szervezetek, gyártók, szoftverházak, nagyon sok kereskedőház és hirdetésekkel foglalkozó intézmény, valamint más számítástechnikai lapok vezetői és munkatársai is.

Csak rövid tájékoztatásként elmondom, hogy a szerkesztőség a lapon kívül számtalan más — a számítástechnikához kapcsolódó — kiadványt is megjelentet: könyveket, speciális szakmai tájékoztató anyagokat, szoftvert árul, és még sok minden mással is foglalkozik. — Egyszer például az egyik korábbi látogatásom alkalmával majdnem lehetetlen volt a lap igazgatójának, Richard Kerler úrnak a szobájába bejutnom, mert minden talpalatnyi helyet IBM-klónokkal teli dobozok foglaltak el.

Az történt ugyanis, hogy a gyártóknak és a forgalmazóknak a lap bejelentette, hogy az NSZK-ban árusított PC-k fontosabb jellemzőit egy új vizsgálati módszerrel fogja megvizsgálni és össze fogja hasonlítani a teljesítményüket. Egy ilyen bejelentésre mindenki, aki a piacon gépügyben érdekelt volt, „összetörte magát", hogy a gépe a tesztelésben részt vegyen. Ha a mi évekkel ezelőtti hasonló akciónkra gondolok, az jut az eszembe, hogy akkor a hazai gyártók és forgalmazók vagy elzárkóztak egy objektív tesztben való részvételtől, vagy befolyásolni igyekezték a lap munkatársait, hogy az eredmények egy kicsit kozmetikázva kerüljenek a lapba. Nem is volt azóta hasonló vállalkozásunk, egy jó kezdeményezés igazságtalan bukása egy időre elegendő tanulságot szolgáltatott számunkra.

Talán ez a példa is érzékelteti, hogy magazinunk más körülmények között dolgozik, mint a CHIP, így feltehetően a Mikroszámítógép Magazin megszületésének ötéves évfordulóján is mellőzzük majd a zajos ünnepségeket, és ha jól sejtem, akkor csak szűkebb — szerkesztőségi körben — fogunk emlékezni az elmúlt évekre. Olvasóink jól tudják, hogy a lap mintegy három hónapos előretartással készül. Most tehát, szeptember elején, amikor ezt a kis eszmefuttatást írom, még nem tudom, valakinek, beleértve volt és jelenlegi gazdáinkat is, eszébe jut-e majd, hogy van és most éppen ötéves a Mikroszámítógép Magazin. A lapot — valamikor 1983 közepén — meglehetősen szokatlan céllal alapítottuk. Egy olyan fórumot akartunk teremteni, amelynek célja a számítástechnikai kultúra tömeges terjesztése, tehát szakmai igénnyel elsősorban a nem számítástechnikai szakemberekhez, az amatőrökhöz és a diákokhoz kívánt szólni. Az elmúlt öt év alatt megtanultuk, hogy az olvasók igénye és érdeklődése igen széles körű, amelyet kiszolgálni egyáltalán nem könnyű, és nem vagyok biztos abban, hogy minden igyekezetünk ellenére sikerült eleget tennünk valamennyi olvasónk várakozásának. Az olvasói levelekből kiderült, hogy nemcsak az amatőrök, de a szakma is olvassa és figyelemmel kíséri lapunkat, és nem csupán a hibáinkra reagálnak gyorsan, de nagyon sok tanácsot is kapunk, hogy hogyan tehetjük színvonalasabbá a magazint.

Annak idején Könyves Tóth Pállal órákat vitatkoztunk kettesben, amig kidolgoztuk a lap tervét és amig az első szerkesztőségi ülésen megalakult az első szerkesztőbizottság: Broczkó Péter, Budai György, Garádi János, Jakab Ágnes, Nacsa Sándor, Pataki Ernő, Petróczy Judit, Pogány Csaba, Simonyi Endre, Szabó János, Varga András, Vass Nándor és Votisky Zsuzsa.

Az eredmény: valamikor 1983 karácsonyára utcára került az első szám, amelyet — nem lévén lapengedélyünk — az NJSZT kiadványaként publikáltunk.

Az első, a „történelmi" szám hozzávetőlegesen tizezer példányban jelent meg. Ma az olvasói kör már stabilizálódott, és a lap általában húszezer körüli példányban kerül az olvasókhoz. Nem titok, hogy amióta megkezdődött a gazdasági „szabályozók" drasztikus változása, valamint az áremelések sorozata a nyomdaiparban, azóta állandó gondunk a lap gazdasági stabilitásának a fenntartása.

Visszatérve a lap történetére, az első két évben jelentős támogatást kaptunk a Tudományszervezési és Informatikai Intézettől — gyakorlatilag az oktatási kormányzattól, hiszen vállaltuk és támogattuk az iskolai számítástechnika és az iskolán kívüli diákképzés ügyét. Ezzel a témával még ma is jelentős terjedelemben foglalkozunk, annak ellenére, hogy évek óta az oktatásügy — többszöri ajánlkozásunk ellenére — nem érezte magáénak a Mikroszámítógép Magazint, és valószínűleg ezért, a támogatását is megszüntette. Szerencsére más a helyzet a tanárokkal és a diákokkal, akiktől folyamatosan nagyon sok levelet, hozzászólást, cikkeket és programokat, no és ötleteket, javaslatokat is kaptunk, nem beszélve az anyagi támogatásukról, hiszen „veszik a lapot".

Volt egy mintegy másfél éves „flörtünk" a KISZ KB-val is, amiért nagyon hálásak vagyunk, hiszen ez alatt az idő alatt jelentős támogatást kaptunk az ifjúsági szervezettől. Sajnos új „félgazdánk"-kal 1988 közepén az együttműködés megszűnt, a szerződést felmondták, és ma is-mét csak egyetlen tulajdonosa van a lapnak, a Neumann János Számítógéptudományi Társaság, támogatásuk nélkül a lapot nem lehetne megjelentetni.

Nagyon sok fórumon javasolták az elmúlt években, hogy emeljük a lap árát, legalábbis „bruttósítsuk" a hosszú ideje 30 Ft-os árat. Mindeddig ellenálltunk a nyomásnak, aminek persze a jelenlegi érvényes és sokak által kritizált postai terjesztési szabályozás is az oka. A mai rendelkezések szerint ugyanis a posta a lap árának az egyharmadát elviszi terjesztési költségek címén. A saját terjesztés összehasonlíthatatlanul kevesebbe kerülne, mint a jelenlegi 10 Ft-os példányonkénti költség. Ha például a lap árát a háromszorosára emelnénk, akkor ugyanezért a mai „szolgáltatásért", tehát ugyanannyi postai munkáért háromszor annyit kellene fizetnünk, ami jó a postának, és rossz mindenki másnak. Aztán azért sem emeltünk árat, mert akkor — véleményünk szerint — jelentősen csökkent volna a lap iránt az érdeklődés. A mai kereseti és adóviszonyok mellett a családi háztartások minden fillér többletkiadásra nagyon érzékenyek, a költségeket még egyáltalán nem fedező, például 60 Ft-os példányonkénti ár negatív következményei — véleményem szerint — beláthatatlanok lennének. Sajnos a jelenlegi ár már sokáig nem tartható, nagyon valószínű, hogy a jövő évben a lap árát — követve a pénz romlását — mérsékelten emelnünk kell.

Tudvalevő, hogy az anyagilag sikeres lapokat a hirdetések tartják el. Sajnos egyre kevesebb a hal, ugyanakkor a fókák száma növekedik, így sokan tartanak igényt a hirdetők támogatására. Mi ezen a téren csak néha voltunk ügyesek, az olvasók akkor is elverték rajtunk a port, mondván, hogy „hülye hirdetésekre" pazaroljuk az érdekes írásoknak fenntartott oldalakat. (Idézet a diákok leveleiből.)

Most az írás végén — érzem — egy nagyon optimista, lendületes befejezés kellene, nagyjából az ötvenes évek szocreál újságíró-ábrázolásainak megfelelően: egyik kezemben a lap, a másikban egy írásra kész golyóstoll, és egy kissé felemelt fejjel fürkészem a végtelent. Ami a lap tartalmát illeti, azt hiszem valóban nagyon optimisták lehetünk. Nem hiszem, hogy csak a barátaink ragadtak volna tollat és írták meg a véleményüket a lapunknak. S talán az olvasó azt sem feltételezi rólam, hogy kizárólag a jó véleményeket szoktam az „Olvasó írja" rovatban közölni, ám olvasóink döntő többsége valóban elégedett a cikkek színvonalával és a lap szerkesztésével is — azt írják, hogy nem vagyunk unalmasak. Észrevették, hogy fejlődünk, szívesen fogadják, hogy igyekszünk sok minden újat bemutatni. Nagyon jó munkatársaink vannak, dolgozik és igen aktív a diákszerkesztőség (a Csipegető rovat), azt hiszem, nem élünk távol a való világtól, és — hogy egy közhelyet is mondjak — rajta a kezünk a magyar számítástechnika ütőerén. Megpróbálunk gazdaságpolitizálni is — egyelőre kevés sikerrel. Szóval, ha rajtunk múlik, akkor a fejlődésünk töretlen lesz, persze csak azt igérhetjük, hogy továbbra is frissek és érdekesek igyekszünk maradni.

Ami pedig az anyagi hátteret illeti, arra vonatkozóan is megvannak az elképzeléseink. Az írást a CHIP magazin tizéves évfordulójával kezdtem, nem véletlenül. Most éppen a közepén vagyunk egy tárgyalássorozatnak, hogy a CHIP-pel közös lapot és közös kiadóvállalatot hozzunk létre. Ma, szeptember elején még nem tudjuk, mit hoz a jövő, nagyon reméljük, hogy kedvezően zárulnak a megbeszélések, és a Vogel hathatós anyagi és erkölcsi támogatásával, s persze hazai gazdáinkkal sikerül újjászületnünk.

Én ezt kívánom mindnyájunknak a Mikroszámítógép Magazin ötödik születésnapján. Olvasóinknak pedig — a szerkesztőség valamennyi munkatársa nevében — kellemes karácsonyt és sikeres, boldog új évet!

KOVÁCS GYŐZŐ

STATISZTIKAI PRÓBÁK PROGRAMOZÁSA

A matematika alkalmazási területei közül talán a statisztikai feladatok igénylik a legtöbb numerikus számolást. Legyen szó egyszerűbb leíró statisztikai feladatokról, mint például átlag- és szórásszámításról vagy a nagyobb elméleti apparátust igénylő matematikai statisztikáról. Nem véletlen tehát, hogy már a számítógépek megjelenése előtti időben is törekedtek a nagyszámú adattal végzendő számítások gépesítésére, azok legalább részbeni automatizálására. Ezzel magyarázható az is, hogy a legegyszerűbb házi számítógépektől a mamutrendszerekig minden géptipushoz a szoftverházak kínálatában a statisztikai feladatok felhasználói programjainak gazdag választékát találjuk.

Az viszont felettébb meglepő, hogy ezek között a matematikai statisztika körébe tartozó programok szinte kivétel nélkül csonkák, hiányosan oldják meg a feladatot. Mind a programcsomagok, mind a szubrutinkönyvtárak a szóban forgó feladatokra olyan komplex programokat vagy programok építésére alkalmas modulokat tartalmaznak, amelyek tökéletesen másolják azt a technológiát, melyet a hagyományos számológészközökre támaszkodó statisztikai gyakorlat az „őskorban" alkalmazott. Régóta keresem a változást, az előrelépést a programozási felfogásban, de annak még kísérletével sem találkoztam. Lehet, hogy léteznek teljes megoldások, a hiányosakból viszont annyi van, hogy minden bizonnyal nem lesz hiábavaló az alábbi elemzés.

A kifogásaimat, helyesebben aggályaimat egyrészt széles körben alkalmazott statisztikai vizsgálat, a khi-négyzet analízis programozásának részletes tervezésével illusztrálom, amivel talán azt is sikerül bemutatnom, hogy gyakran egy „kis program" tervezése is milyen gondos és alapos elemzést igényel.

A khi-négyzet próba

A teljességre törekvés nélkül, röviden tekintsük át a statisztikai próbák elvi alapjait és kivitelezésük módját. A vizsgálatot végző kutató gyakran valamilyen tulajdonság meglétére vagy éppenséggel hiányára gyanakszik. Sokszor, például akkor, ha az adatokat ellenőrizhetetlen és kiküszöbölhetetlen mérési hibák terhelik, a kérdésre nem lehet határozott igen/nem választ adni. Ilyenkor konstruálni kell egy olyan, az adatokból — mintából — kiszámítható mennyiséget, az ún. próbafüggvényt vagy statisztikát, melyről tudjuk, hogy a mintaelemek meghatározott kezdeti feltételeinél hogyan viselkedik.

Úgy gondolom, nem lesz haszontalan a gondolatmenet követésére egy konkrét példát elemezni: tegyük fel, hogy ellenőrizni akarjuk számítógépünk véletlenszám-generátorát. A kérdés az, hogy a generátor egyenletes eloszlású számsorozatot állít-e elő. Egyszerű programmal meggyőződhetünk arról, hogy egy kísérleti RND sorozat (minta) elemeinek mennyi az átlaga és a szórása. Ha a generátor jól működik, akkor a (0,1) intervallumból vett minta átlaga 0,5, szórása $1/\sqrt{12} = 0{,}288675\ldots$ körül lesz. Ha ettől eltér a kísérleti eredmény, akkor a generátor BIZTOSAN rossz. Ha viszont a két érték az elméletileg várható értékek közelében van, akkor csak annyit mondhatunk, hogy LEHET jó a generátor, a kísérlet nem zárja ki a helyes működést. A megbízható diagnózishoz azonban komolyabb próbára van szükség.

Mikor mondhatjuk azt, hogy a véletlen számsorozat egyenletes eloszlású? Akkor, ha a teljes (0,1) intervallum minden részintervallumába eső elemek gyakorisága az intervallumok hosszával arányos. A kísérletet tehát úgy kell megszervezni, hogy a (0,1) intervallumot tetszőlegesen felosztjuk és megszámláljuk, hogy az előállított RND sorozatnak egy-egy részbe, cellába hány eleme kerül.

Mit várhatunk ettől a kísérlettől? Egyenletes felosztásnál minden cellába ugyanannyi elem fog esni? Majdnem biztos, hogy nem. Azt viszont elvárhatjuk egy „jó" generátortól, hogy KÖZEL egyenlő gyakorságokat produkáljon.

A következő kérdés az, hogy az elméletileg várt egyenletes gyakoriságtól való eltérést hogyan mérjük. Jelöljük f(i)-vel az i-edik cellában mért gyakoriságot és legyen p(i) annak a valószínűsége, hogy egy RND-vel generált véletlenszám az i-edik cellába esik. Ha r elemű sorozatot állítunk elő, akkor az i-edik cellában elméletileg várt gyakoriság r × p(i) lesz. A tapasztalati és az elméleti eloszlásnak az egész (0,1) intervallumra vonatkoztatott eltérését jellemezhetjük a következő próbafüggvénnyel:

$$(1)\ldots \qquad \chi^2 = \sum_{i=1}^{m} \frac{(f(i)-r.p(i))^2}{r.p(i)}$$

ahol m a cellák száma.

Ha egyenletes felosztást alkalmazunk és a példánkban egyenletes eloszlású generátort vizsgálunk, a khi-négyzet statisztika egyszerűbb alakot vesz fel:

$$(2)\ldots \qquad \chi^2 = \frac{m}{r} \sum_{i=1}^{m} (f(i)-r/m)^2$$

Felmerülhet az olvasóban az, hogy miért pont ezt a kifejezést használjuk az eltérések mérésére. Látható, hogy az egyes cellákon belüli eltérések négyzetével számolunk, s ennek az a jelentősége, hogy ily módon az ellentétes előjelű eltérések nem közömbösítik egymást. Az igazi ok persze nem ez, hanem az, hogy az így számított próbafüggvénynek ismerjük a viselkedését. Bebizonyították, hogy ez a statisztika n = m − 1 szabadságfokú khi-négyzet eloszlást követ. Ezeknek a valószínűségi változóknak a sűrűség- és eloszlásfüggvényei a normális eloszlásból levezethetőek:

$$(3)\ldots \qquad k_n(x) \to \frac{x^{(n/2)-1} \cdot \exp(-x/2)}{2^{n/2} \cdot \Gamma(n/2)} \quad ;x>0$$

$$(4)\ldots \qquad p(n,x) = P(\xi < x) = k_n(x) = \int_0^x k_n$$

Térjünk vissza a próba menetére. Mint megállapítottuk, a kísérlet várható eredménye az, hogy a mintából számított (1), illetve (2) próba 0-tól különbözni fog ugyan, de ha igaz, hogy a gép RND függvénye egyenletes eloszlást generál, akkor az eltérés nem lesz túl nagy. De vajon mi az a kritikus érték, amely felett el kell vetnünk az egyenletes eloszlás hipotézisét?

Nyilván akkor kellett így ítélnünk, ha a mintából számított statisztika „valószínűtlenül" nagy. Olyan nagy, amit véletlen mintából soha, de legalábbis nagyon kis valószínűséggel kaphatnánk. Ez utóbbi valószínűség a (4)-ben szereplő esemény komplementerének valószínűsége: 1 − p(n,x).

A statisztikusok 1 − p(n,x)-re a 0,001, 0,01, illetve a 0,05 szinteket használják a leggyakrabban.

A próba algoritmusa

Az elmondottakból következik a próba végrehajtásának menete:

1. Adatgyűjtés (példánkban az RND sorozat képzése).
2. Az f(i) gyakorisági sor előállítása.
3. A (2) próbafüggvény kiszámítása.
4. A (4) valószínűség kiszámítása.
5. Döntés a kiszámított 1 − p alapján.

Ezzel szemben a bírált programok beérik azzal, hogy kiszámítják a próbafüggvényt. (Nemcsak a khi-négyzet próbánál, hanem minden másiknál is.) Az algoritmus 4. lépését egyszerűen elhagyják és a felhasználóra bízzák. Ugyanúgy, mint a hagyományos technológiánál, a felhasználó előveszi a könyvespolcáról a kézikönyvet, függvénytáblát, majd a megfelelő táblázatból kikeresi a döntéshez szükséges kritikus értéket. (Közben lelkiismeretesen interpolál.)

Talán a statisztikai programok készítői úgy gondolták (gondolják), hogy örüljön a t. felhasználó, ha az adatrendezés és a számolási munka egy részétől megkímélik. Arra nem is merek gyanakodni (?), hogy nem tudják megírni a 4. lépést megvalósító programot minden próbához.

Vizsgáljuk meg a khi-négyzet próbánál ezt a kritikus 4. lépést, melyről látni fogjuk, hogy valóban nem egyszerű, de nem is kivihetetlen.

STATISZTIKAI PRÓBÁK PROGRAMOZÁSA

A sűrűségfüggvény vizsgálata

Először foglalkozzunk a (3)-ban szereplő függvények nevezőjében álló

(5)... $c_n = 2^{n/2} \cdot \Gamma(n/2)$

együtthatóval. Az első tényező, a 2-hatvány kiszámítására minden magas szintű programozási nyelvben találunk direkt vagy indirekt műveletet. A második tényezőben álló gamma függvénnyel más a helyzet. Ennek helyettesítési értékét a

(6)... $\Gamma(x+1) = \int_0^{\infty} t^x \cdot \exp(-t)\, dt$

improprius integrál definiálja, amit közelítő formulákkal kívánt pontossággal ki lehetne számítani. Erre azonban nincs szükség, mert az (5) együtthatóban csak speciális argumentumok szerepelnek. Nevezetesen az n szabadsági fok párosságától függően a kiszámítandó helyettesítési érték

(7)... $\Gamma(n/2) = \begin{cases} \Gamma(k) & ; n=2k \\ \Gamma(k+1/2) & ; n=2k+1 \end{cases}$

Ezeket a kifejezéseket a gamma függvény következő tulajdonságainak a felhasználásával meghatározhatjuk:

(8)... $\Gamma(x+1) = x \cdot \Gamma(x)$; x egész

$\Gamma(x+1) = x!$; $\forall x$ egész

$\Gamma(1/2) = \sqrt{\pi}$

Ezzel az (5) együttható egyszerűbb alakban írható fel páros és páratlan n-re egyaránt:

(9)... $c_{2k} = 2^k \cdot (k-1)! = 2 \cdot (2k-2)!!$

$c_{2k+1} = \sqrt{2\pi} \cdot (2k-1)!!$

ahol $n!! = n \times (n-2)(n-4) \ldots (n-2j)$, minden $j:(n-2j) > 0$.

E két esetet egyetlen formulával írhatjuk fel minden n-re:

(10)... $c_n = c_* \cdot (n-2)!!$

ahol páros n-re $c_* = 2$, páratlanra pedig $\sqrt{2\pi}$.

A számításhoz felhasználható a következő rekurzió is:

(11)... $c_1 = \sqrt{2\pi}$; $c_2 = 2$;

$c_{n+2} = n \cdot c_n = c_* \cdot n!!$

Az együttható tehát kiszámítható, csupán az a kérdés, hogy a lebegőpontos aritmetikában nem okoz-e túlcsordulást ez a faktoriális típusú kifejezés. Sajnos 8 bites karakterisztika mellett már n = 58-ra sem végezhető el a számolás (hozzávetőlegesen 1.7E + 38), ezért tovább kell folytatnunk a probléma elemzését.

Vizsgáljuk meg előbb magát az integrandust, de figyelmünket forditsuk az együtthatót nem tartalmazó

(12)... $k_n^* := x \to c_n \cdot k_n(x) = x^{(n/2)-1} \cdot \exp(-x/2)$

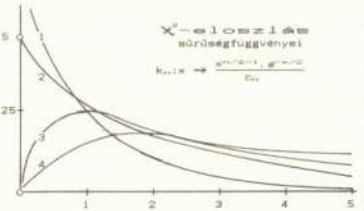
függvényekre. Ezek menetét az ismert analitikus módszerekkel megállapithatjuk. Mint az a grafikonokon *(1. ábra)* is látható, a sűrűségfüggvények három típust képviselnek:

— Az n = 1 esetben a 0 jobb oldali környezetében a függvény nem korlátos, tehát a (4) integrál improprius.

— Az n = 2 esetben 0-nál a jobb oldali határérték 0,5, a függvény monoton csökken.

— Minden n > 2-re a sűrűségfüggvényeknek x = n − 2 helyen maximumuk van. Az x = 0-nál a határérték 0.

A három típusban közös, hogy + ∞-ben 0 a határérték.



1. ábra

Az eloszlásfüggvények

A szignifikancia szintjének meghatározásához, azaz a (4) határozott integrál kiszámításához ismernünk kell a sűrűségfüggvények primitív függvényeit. Célszerűbb az együtthatót nem tartalmazó (12) függvényekkel foglalkozni. Az $u = \exp(-x/2)$ és $v' = x \uparrow (n/2-1)$ választással, parciális integrálással azt kapjuk, hogy

(13)... $\int k_n^* = \frac{2}{n} \cdot x^{n/2} \cdot \exp(-x/2) + \frac{1}{n} \int k_{n+2}^*$

Ebből rekurzív helyettesítéssel és azonos átalakitással, majd az eredeti sűrűségfüggvényekre visszatérve, azok primitív függvényeit a

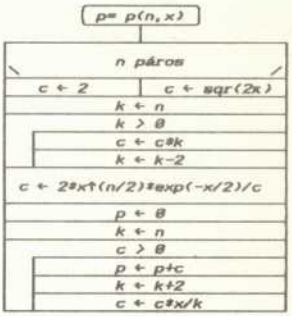
(14)... $\int k_n = \frac{2x^{n/2} \cdot \exp(-x/2)}{n \cdot c_n} \cdot \left(1 + \frac{x}{n+2} + \frac{x^2}{(n+2)(n+4)} + \ldots\right)$

függvénysorral állithatjuk elő. Mivel ezek helyettesítési értéke az x = 0 helyen zérus, a keresett (4) valószínűségeket e függvénysorok x = khi↑2 helyen vett helyettesítési értéke adja. Ezt természetesen nem lehet „pontosan" kiszámítani, de látható, hogy a zárójelben álló sor tagjaiban a nevező igen gyorsan növekszik, s így a konvergencia is gyors lesz.

A számítás algoritmusa

A p(n,x) = P(ξ < x) valószínűség kiszámítására a (11) rekurzióból és a (14) függvénysor kiszámításából álló egyszerű programot vezethetünk le.

A program elején az n párosságától függően állítjuk be a (11) rekurzió kezdőértékét (c:= . . .). A rekurzió végén kiszámítjuk a (14) zárójelén kívül álló tényezőt, melyet a c változóban viszünk tovább. A függvény kiszámítható egyszerű összegzéssel, az összeget a p változóban képezzük. Az összegző ciklust az aktuális tag alácsordulásával állítjuk le. A programban szereplő k változó a ciklusok futó paramétere.



A túlcsordulás kivédése

A kiszámítandó p(n,x) valószínűség a definícióból következően mindig a (0,1) intervallumba esik, ha x > 0. Arról kell tehát gondoskodnunk, hogy a számítási algoritmus közbenső eredményei ne lépjenek ki a programot futtató számítógép lebegőpontos formátumának korlátai közül. Sok esetben megoldást nyújt a sorozattal számított mennyiségek logaritmikus kiszámítása, természetesen csak akkor, ha a logaritmikus számításból való visszatérést az algoritmusnak azon a pontján végezzük el, ahol a számított mennyiség már az ábrázolható tartományba esik. A p(n,x)-re írt algoritmust a példánkhoz készített BASIC programban ennek megfelelően úgy módosítottuk, hogy a c iterációjában használtunk logaritmikus transzformációt, s a visszatérést akkor végeztük, amikor a (14) zárójelén kívüli helyettesítési értéket kiszámítottuk. Belátható, hogy ennek a számításnak az eredménye mindig kisebb 1-nél, hiszen a másik tényező pozitív x-re mindig 1-nél nagyobb.

Nem állítjuk, hogy a túlcsordulás lehetetlen, de csak akkor következik be, ha a közbenső értékek logaritmusa ábrázolhatatlan. Ez azonban csak irreálisan nagy, a gyakorlatban elő nem forduló szabadsági fokoknál következne be.

Megtörténhet azonban, hogy nagy x mellett a (14) elején álló kifejezés alácsordul, és p=0 eredményt ad. Ezt az extra esetet van hivatva kivédeni a szubrutin 1180-as sorában szereplő vizsgálat.

Látható, hogy a program kódolásakor a függvénysor iterációjának ciklusát a c>0,0000001 feltétellel vezéreljük. Ezzel is kielégítő pontosságot érhetünk el.

A szubrutin alkalmazása

A működés bemutatására a szubrutint a példaként szereplő RND-teszthez készített keretprogrammal futtattuk. Ennek listájából a kiírást végző modult elhagytuk. Érdemes megfigyelni, hogy bár mindkét kísérletben *(2. ábra)* a próba az elfogadási tartományban van (1−p>5%), a jobb illeszkedést mutató 2000-es mintában az átlag és a szórás jobban eltér az elméleti értéktől.

```
100 REM **********************
110 REM *      RND-TEST        *
120 REM *  BASIC : C8M V3.5 *
130 REM **********************
140 PRINT"■■■■  RND-GENERATOR TESZT"
150 INPUT "■ KISERLETEK SZAMA=";R
160 INPUT "■     CELLAK SZAMA=";M
170 IF R<10*M THEN 150
180 :
190 TI$="000000"
200 DIM F(M)
210 REM --- GYAKORISAGI TABLA ---
220 A=0 : S=0
230 FOR I=1 TO R
240 : F=RND(0)
250 : A=A+F
260 : S=S+F*F
270 : J=INT(M*F)+1
280 : F(J)=F(J)+1
290 NEXT I
300 A=A/R              : REM ATLAG
310 S=SQR(S/R-A*A) : REM SZORAS
320 :
330 REM --- KHI↑2 STATISZTIKA ---
340 X=0
350 FOR J=1 TO M
360 : X=X+(F(J)-R/M)↑2/R*M
370 NEXT J
380 N=M-1
390 IF X=0 THEN P=0 : GOTO 430
400 ::::::::::::::::::::::::::::::::::
410 :    GOSUB 1000: REM P ← N,X   :
420 ::::::::::::::::::::::::::::::::::
430 TM$=TI$   : REM TISZTA FUTASIDO
440 :
450 REM ---   OUTPUT ------------
  :
  :
999 END
1000 REM =====  P ← N,X  =====
1010 REM X::= KHI↑2 PROBAFUGGVENY
1020 REM N::= SZABADSAGFOK
1030 IF X=0 THEN P=0 : GOTO 1190
1040 IF N/2=INT(N/2) THEN C=2 :ELSE
                                C=SQR(2*x)
1050 K=N   : C=LOG(C)
1060 DO WHILE K>0
1070 : C=C+LOG(K)
1080 : K=K-2
1090 LOOP
1100 C=2*EXP((N*LOG(X)-X)/2-C)
1110 P=0
1120 K=N
1130 DO WHILE C>0.0000001
1140 : P=P+C
1150 : K=K+2
1160 : C=C*X/K
1170 LOOP
1180 IF P=0 AND X>N THEN P=1 :REM EXTRA
1190 REM P::= P(N,X) SZIGNIFIKANCIA
1200 RETURN
```

RND-TESZT: COMMODORE- PLUS/4

i	f(i)	Hisztogram
1	194	
2	201	
3	187	
4	190	
5	189	
6	218	
7	212	
8	208	
9	205	
10	196	

```
Kisérletek sz.=  2000
Szabadsági fok=     9

     Khi↑2   =  5.0000
  p(n,khi↑2)=  0.1657
     1-p     =  0.8343

     Átlag   =  0.5062
  Várt átlag =  0.5000

     Szórás  =  0.2873
  Várt szórás=  0.2887

   Futási idő :  00:01:04
```

i	f(i)	Hisztogram
1	4945	
2	4891	
3	5048	
4	4985	
5	5071	
6	4963	
7	5059	
8	5113	
9	5000	
10	4925	

```
Kisérletek sz.= 50000
Szabadsági fok=     9

     Khi↑2   =  9.1448
  p(n,khi↑2)=  0.5759
     1-p     =  0.4241

     Átlag   =  0.5010
  Várt átlag =  0.5000

     Szórás  =  0.2878
  Várt szórás=  0.2887

   Futási idő :  00:26:38
```

2. ábra

LOTTÓ-TESZT:

3. ábra

1	2	3	4	5	6	7	8	9	10
79	68	114	84	66	86	89	85	79	99
11	12	13	14	15	16	17	18	19	20
67	98	99	88	88	74	80	99	91	84
21	22	23	24	25	26	27	28	29	30
79	85	96	88	89	69	66	75	100	62
31	32	33	34	35	36	37	38	39	40
71	80	78	94	87	83	77	82	70	74
41	42	43	44	45	46	47	48	49	50
89	94	79	78	85	89	96	78	95	80
51	52	53	54	55	56	57	58	59	60
95	76	83	81	77	105	73	69	77	91
61	62	63	64	65	66	67	68	69	70
74	75	67	93	84	85	93	72	93	73
71	72	73	74	75	76	77	78	79	80
86	83	83	78	108	79	96	82	77	71
81	82	83	84	85	86	87	88	89	90
85	75	82	96	78	101	85	67	82	85

Érdeklődésre tarthat számot egy másik próba, mellyel a lottó számsorsolásának egyenletes eloszlását vizsgáltuk *(3. ábra)* az első 1500 játékhét (1985. II. 29.) után. Az egyenletes eloszlástól való eltérés gyengén szignifikáns: 1−p=3%. Meg kell jegyezni, hogy ugyanez a próba korábban még jó illeszkedést mutatott.

Dr. Hack Frigyes

```
Húzások száma =  1500
Szabadsági fok=    89

     Khi↑2   = 115.6080
  p(n,khi↑2)=  0.9695
     1-p     =  0.0305
Elméleti gyak.=  83.3333
```

FELADATOK — MEGOLDÁSOK

Sorozatunkat elsősorban középiskolásoknak szánjuk, de reméljük, hogy minden olvasónknak tanulási lehetőséget és szórakozást nyújt.

A feladatok a Nemes Tihamér országos számítástechnikai verseny színvonalának felelnek meg. Minden esetben olyat választunk, amely röviden, gyorsan megoldható, de megoldásához ötletre van szükség. A megoldást mindig a következő számban közöljük.

Mivel változatosságra törekszünk, különböző programozási nyelveket használunk. Az is előfordul majd, hogy egy feladatra több programnyelven is közlünk megoldást, ezzel is elősegítve az ismeretszerzést.

A szerkesztőség várja az olvasók, a versenyzők leveleit. A legötletesebb program beküldőjét könyvutalvánnyal jutalmazzuk. Ne feledjenek azonban a programhoz leírást is mellékelni!

6. feladat: Osztás

Írjon programot, amely egy nyolcvanjegyű előjeles egész számot eloszt egy negyvenjegyűvel, majd kiírja a hányadost és a maradékot!

A megoldás

A feladat jellegéből kínálkozik, hogy a programot assembly nyelven írjuk meg. Ez különösebb nehézség nélkül megoldható, és gyors, hatékony kódot eredményez. Az előző feladat, a szorzás, a mostanihoz hasonló, amiért is a megoldásának programnyelve a 8086 assembly.

De mit tegyen az az olvasó, aki az assembly programozásban kevésbé jártas? Megoldható a feladat más nyelven is? Természetesen igen, akár BASIC-ben is. Az alább ismertetett programmal is ezt igazolom. Kétségtelen, hogy ez nem egy jól használható, hatékony program, ám ezzel a feladat tökéletesen megoldható.

Azzal, hogy a program megírására magas szintű nyelvet választottunk, még nem mentesülünk a 80 jegyű szám ábrázolásának gondjaitól. Semelyik egyszerű változó nem alkalmas ilyen méretű számok ábrázolására, ezért valamilyen összetett adattípust kell létrehoznunk. Ez célszerűen valamilyen tömb lesz.

Valós számokból álló tömb használatakor tudnánk egy elemen a legtöbb jegyet tárolni. Ekkor azonban a valós szám ábrázolását az adott gépen meg kellene ismerni, és még így sem biztos, hogy végeredményben hatékonyabb programot kapunk, a valós számokkal végzett műveletek lassúsága miatt.

Marad tehát az egész és a karakteres ábrázolási mód. Ha a programot BASIC-ben írjuk, akkor a karakteres ábrázolás a célszerűbb, mivel így a BASIC sztringműveleteket használhatjuk. Más magas szintű nyelv alkalmazásánál valószínűleg az egészekből álló tömb lenne előnyösebb.

El kell azt is döntenünk, hogy egy karakteren hány jegyet ábrázolunk. Értelmezhetjük a tömböt egymást követő bájtokként és elhelyezhetjük a számot ezeken bináris alakban. Ekkor a beolvasásnál és a kiíráskor szükséges konverzió bonyolultabb lesz, de cserébe a számítás gyorsabbá válik, és a tárolásra felhasznált terület minimális lesz. (Ehhez hasonló ábrázolást használtunk az előző feladat megoldásánál.)

Más számábrázolás is elképzelhető: tárolhatunk két jegyet egy karakteren BCD formában, vagy akár egy jegyet karakteres formában is. Nyilván ez utóbbi bánik legpazarlóbban a memóriával, de a számok beolvasásánál és kiírásánál minimális átalakítást igényel.

Ez utóbbit használja a mellékelt *program* is. A program ZX-Spectrumra készült, de semmilyen különleges utasítást nem használ, így könnyen átvihető más gépre.

A program elején, a 90—160-as sorokban a változókat definiáljuk. A 180-as sorban látható értékadás dönti el, hogy hány jegyű számokkal végezzük a műveletet. Az osztó és a maradék 41 jegyű, hogy a számításnál fellépő átvitelt is tárolni lehessen. Az e$ az előjelét tárolja. Ez már a beolvasáskor eldől.

A program következő részében (190—290-es sorok) a számokat olvassuk be. A beolvasott sztringet a 2000-es Alakít szubrutin a számításokhoz megfelelő 80 jegyűvé alakítja. Az eredmény előjelét már itt meg lehet határozni. Bár ez a szubrutin a program leghosszabb rutinja, megértéséhez különösebb magyarázat nem szükséges, hiszen csupán egy sor feltételvizsgálatból áll.

A számítás teljes egészében 3000-es Osztás szubrutinra hárul. Az algoritmus majdnem teljesen megegyezik az általános iskolában megismert írásbeli osztáséval.

Először m$-ba rakja az osztandó első jegyét és ebből annyiszor vonja ki az osztót (o$), amíg m$<o$ nem lesz. A hányados első jegyét a levonások száma adja, m$ pedig a maradékot tartalmazza. Ezután m$-hoz hozzáveszi az osztandó következő jegyét és újra elvégzi a kivonásokat. Ezek száma adja a hányados következő jegyét.

Mindez így ismétlődik az osztandó utolsó jegyéig. Ekkor m$-ban az osztás maradéka található, h$-ban pedig már megvan a hányados összes jegye.

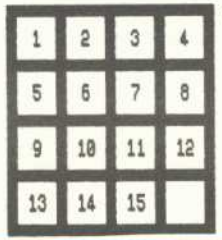
A 0-val való osztás is itt küszöbölhető ki. Ez akkor válik nyilvánvalóvá, amikor már 10 kivonás után sem lehet m$<o$.

A kivonást a 2500-as Kivonás rutin végzi. Az írásbeli kivonásnál is használt algoritmus könnyen bővíthető. Az átvitelt mindig cy tartalmazza.

A program befejező része (600-as sortól) az eredmény kiírása. A kiírás előtti formálás célja a felesleges vezető nullák törlése.

Az új és a következő feladat az egyik legrégibb logikai játékhoz, a tizenötös játékhoz kapcsolódik. A végeredmény egy önállóan is jól használható játékprogram lesz, és a gyűjtők igényeit is kielégíti.

A játékot, amelyet bizonyára mindenki ismer, Sam Loyd találta fel 1878-ban. Megjelenése olyasféle szenzációt keltett, amelyhez csak a bűvös kockáé mérhető. Mindmáig a legelterjedtebb logikai játékok közé tartozik. Ezt bizonyítja a rengeteg műanyagból, fémből, fából készült „manuális" változata. A számítógépes megvalósítás azonban a mai napig késik. Ezt a hiányt kívánja pótolni a következő két feladat.




```
 10 REM ==========================
        80 jegyű szám osztása
          40 jegyűvel.

        1988.07.01.

        Pintér Gábor
 ==========================

 90 REM ---------------------
        Változók definiálása
 ---------------------
 100 LET j40=40
 105 LET j80=j40+j40
 110 DIM o$(j80): REM osztandó
 120 DIM s$(j40+1): REM osztó
 130 DIM m$(j40+1): REM hányados
 140 DIM m$(j40+1): REM maradék
 150 LET e$="": REM előjel
 160 DIM p$(j80): REM 0-k
 162 FOR i=1 TO j80
 164 LET p$(i)="0"
 166 NEXT i
 180 REM ---------------------
        Beolvasás
 ---------------------
 190 PRINT "Osztandó:"
 200 INPUT "Osztandó: "; LINE a$
 205 PRINT a$
 210 GO SUB 2000: REM alakít
 220 IF a$="" THEN GO TO 200
 230 LET o$=a$
 240 PRINT : PRINT "Osztó:"
 250 INPUT "Osztó: "; LINE a$
 260 PRINT a$
 270 GO SUB 2000: REM alakít
 280 IF a$="" THEN GO TO 250
 282 IF a$( TO j40)<>p$( TO j40)
 THEN GO TO 250
 285 LET s$=a$(j40 TO )
 290 PRINT : PRINT
 500 REM ---------------------
        A számítás
 ---------------------
 510 GO SUB 3000: REM Osztás
 600 REM ---------------------
        Kiíratás
 ---------------------
 610 LET a$=h$
 620 GO SUB 3500: REM Kiíratási
 formálás
 625 PRINT "Hányados:"
 630 PRINT a$
 635 PRINT
 640 LET a$=m$
 650 GO SUB 3500: REM Kiíratási
 formálás
 660 PRINT "Maradék:"
 670 PRINT a$
 1000 STOP
 2000 REM ---------------------
        Alakít
 ---------------------
```

A játék 15 számozott, négyzet alakú lemezkéből áll, amelyek szabadon csúszkálnak egy négyzet alakú dobozban. A dobozban 4×4=16 hely van, ezért egy hely mindig szabadon marad. A játék célja egy összekevert helyzetből az *ábrán* látható kiindulási állapot visszaállítása. Az elrendezést úgy kell elérni, hogy a lemezeket csak csúsztatni szabad, a dobozból kiemelni nem.

A játék sajátossága, hogy nem minden összekevert helyzetből rakható ki.

7. feladat: Tizenötös játék 1.

Írjon programot, amely kirajzolja a tizenötös játékot, lehetővé teszi a számok megfelelő mozgatását és össze tudja keverni a játékot! Ügyeljen arra, hogy az összekevert állás kirakható legyen!

PINTÉR GÁBOR

```
2010 REM Értelmezi a beolvasott
karaktereket.
2020 REM Ha értelmes, az előjele
t levégja és a számot 80 jegyűvé
alakítja.
2030 REM Ha nem értelmes a$="" i
s lesz.
2040 REM Előjelet állítja.
2045 IF a$="" THEN RETURN
2050 LET i=0
2055 LET f$=a$: REM A beolvasott
előjel.
2060 LET i=i+1
2070 IF a$(i)="+" THEN GO TO 212
0
2080 IF a$(i)<>"-" THEN GO TO 21
00
2090 IF a$="" THEN LET f$="-": G
O TO 2120
2100 LET f$="": GO TO 2120
2110 IF a$(i)>="0" AND a$(i)<="9
" THEN LET a$(i)=i: GO TO 2190: R
EM Az első számjegy.
2120 IF i<LEN a$ THEN GO TO 2060
2130 REM A sorban nincs számjegy
.
2140 LET a$=""
2150 RETURN
2170 LET i=i+1
2180 IF a$(i)<"0" OR a$(i)>"9" T
HEN LET i=i-1: GO TO 2200
2190 IF i<LEN a$ THEN GO TO 2170
2200 REM A szám vége.
2205 IF i-elso)=80 THEN GO TO 21
50
2210 LET a$=p$( TO j80-1-i+elso)
+a$(elso TO i)
2220 LET a$=f$
2230 RETURN
2500 REM ---------------------
        Kivonás
 ---------------------
2510 REM Kivonja m$-ból s$-et.
2520 REM Eredmény m$-be kerül.
2530 LET cy=0
2540 FOR i=j40+1 TO 1 STEP -1
2550 LET sz=CODE m$(i)-CODE s$(i
)-cy
2560 LET cy=sz<0
2570 IF cy THEN LET sz=sz+10
2580 LET m$(i)=CHR$ (sz+48)
2590 NEXT i
2600 RETURN
3000 REM ---------------------
        Osztás
 ---------------------
3010 REM Elosztja o$-el s$-el.
3020 REM Az eredmény h$-be, a ma
radék m$-be kerül.
3030 REM Ha s$=0: hibajelzéssel
leáll.
3040 LET m$=p$( TO j40+1)
3050 FOR i=1 TO j80
3060 LET m$=m$(2 TO )+o$(k)
3070 LET er=48: REM =CODE "0"
3080 IF m$<s$ THEN LET h$(k)=CHR
$ er: NEXT k: RETURN
3090 GO SUB 2500: REM Kivon
3100 LET er=er+1
3110 IF er<=57 THEN GO TO 3080
3220 PRINT "0-val való osztás ér
telmetlen!"
3230 STOP
3500 REM ---------------------
        Kiíráshoz formálás
 ---------------------
3510 REM a$-ból törli a vezető 0
-kat és hozzátesszi az előjelet.
3520 LET i=0
3525 LET i=i+1
3530 IF a$(i)<>"0" THEN GO TO 35
80
3540 IF i<LEN a$ THEN GO TO 3525
3550 REM A szám 0.
3560 LET a$="0"
3570 RETURN
3580 LET a$=e$+a$(i TO )
3590 RETURN
```

Digitális montázskészítő

```
10 REM Digitalis montazskeszito by
       Szarka Zoltan
20 Cls
30 Input "Milyen hosszusagu montazst
       kivan betolteni(max25sec)";A
40 If A<=0 or A>25 then 20 Else A=INT(
   A)
50 Print "1 sec kb 1250 byte tart
   foglal el.Tehat az On altal kert
   ";A;" sec ";A*1250;" byteot igenyel
   !"
60 Print "Ez megfelel Onnek (I/N)?"
70 If Inkey$="I" Then 80 Else If inkey
   $="N" Then 20 Else 70
80 CM=58880-A*1251 : If CM<18000 Then
   Print "Nem fer el a tarban !" :End
   Else X=CM-INT(CM/256)*256:Y=INT(CM
   /256):W=A*1250-INT((A*1250)/256)*
   256:Z=INT((A*1250)/256)
90 Poke20000,33,59,64,62,127,119,211,0
       ,33,x,y,17,w,z,243,6,8,175,219,
   0,31,31,31,203,17,197,6,10,0,16
   ,253,193,16,239,113,35,27,122,1
   79,32,230,251,201
100 Print"Figyelem allitsa a magneto-
   font lejatszasra es a RETURN le-
   nyomasaval egyidoben inditsa is
   el!"
110 Print "(RETURN)"
120 If Inkey$<>CHR$(13) Then 120 Else
   Beep 33,33
130 C=CALL(20000) : Print "A beolvasas
   megtortent!"
140 CM=CM-150 : If CM>=32768 Then CM=
   CM-65535
145 Print "Az inditasi cim:";Print tab
   (10);CM
150 PokeCM,33,x,y,17,w,z,243,78,6,8,20
   3,17,159,197,6,10,0,16,253,193,254
   ,255,40,15,62,45,211,0,16,236,35,2
   7, 122,179,32,227,251,24,13,62,63,
   211,0,16,221,35,27,122,179,32,212,
   251,62
160 Poke CM+53,238,33,59,64,119,211,0,
   201
170 Print "Nyomja meg az 'I'-t es a
   montazs megszolal!"
180 If Inkey$<>"I" then 180 Else C=
   CALL(CM)
190 Print "Ez megfelel (I/N)?"
200 If Inkey$="I" Then 210 Else If
   Inkey$<>"N" Then 200 Else Print"
   Tisztabb felvetellel probalkozz!"
   : Goto 30
210 Input"Milyen neven mentsem";A$:
   HQ=LEN(A$) : If HQ>16 Then HQ=16
220 Print"Figyelem kimenten kovetkezik
   !":Print"Allitsa felvetelre a mag-
   not, ha kesz,nyomja meg a RETURN-t
   es inditsa el!"
230 Print "(RETURN)"
240 If Inkey$<>CHR$(13) Then 240 Else
   Beep 33,33
250 If CM<>ABS(CM) Then CM=CM+65535
255 E=CM-INT(CM/256)*256:K=INT(CM/256)
260 Poke24000,33,92,64,62,131,119,33,2
   16,89,205,146,0,33,94,64,62,1,119,
   43,43,62,249,119,33,0,230,1,0,0,17
   ,e,r,205,149,0,33,94,64,126,60,39,
   119,43,43,62,177,119,205,155,0,201
270 Poke23000,HQ,219,89 :FOR I=1 TO HQ
   : Poke23003-1+I,ASC(MID$(A$,I,1)):
   Next I
280 Q=CALL(24000)
290 END
```

A magazin 1988/7. számában megjelent Hangdigitalizálás programon belüli alkalmazását szemlélteti ez a rövid példaprogram. Akik kipróbálták a digitalizáló programot, bizonyára észrevették, hogy a hangokat zajosan kapták vissza. Ennek oka a szalag és a felvétel minősége, valamint a kis piezoelektromos „hangszóró". A montázst készítő programnál ezt figyelembe kell venniük, tehát lehetőség szerint jó minőségű felvételt játsszunk be. Legtisztábban szintetizátor-, gitár- és más számítógéphangokat kaphatunk vissza.

Ennek a programnak a segítségével adott időtartamú hanganyagot tölthetünk számítógépünkbe, majd ezt fájlként magnóra vehetjük. Ezáltal lehetőség kínálkozik arra, hogy valamelyik programunkba töltsük és használjuk. A fájlt LOAD utasítással olvashatjuk be.

Figyelem! Minden egyes másodperc 4,8 blokknak felel meg! A program 64-es gépre készült a nagy memóriaigény miatt. A másodpercek időtartama csak a gyári frekvencián futó gépeknél érvényes. Működéskor nagybetűket használjunk és ügyeljünk a számok pontos begépelésére is.

Hogyan indíthatjuk el az elkészült és kimentett montázst? A program futásakor kiírja az indítási címet, ezt jegyezzük fel, mert a később betöltendő montázst ezen a címen kell meghívnunk. A hangfájl használata tehát a következő:

— Elkészítjük a felhasználóprogramot, ügyelve arra, hogy a fájl ne jusson a memória területére (ami a képernyő-memória előtti másodpercszer 1250 bájtot jelent).

— A programunk elején LOAD-dal betöltjük a fájlt.

— Szükség esetén a következő utasítással szólaltathatjuk meg: ?CAU (feljegyzett cím) vagy A=CAU (feljegyzett cím).

SZARKA ZOLTÁN

RESET

SAJÁTKEZŰLEG

Az Egy gép száz bajt csinál? című riportban (1988/11. szám) szó volt a TVC-vel kapcsolatban a RESET gomb hibájáról, amelynek orvoslását — a konstrukció átalakítását — a gyár nem vállalja.

A lényeg az, hogy a fent említett kapcsoló egyszeri lenyomása meleg, kétszeri lenyomása hideg RESET-et eredményez. Ezen házilag elég könnyű segíteni. Miután kéznél van egy pillanatkapcsoló, egy vékonyabb vezeték és egy páka, leszereljük a számítógép hátlapját. Itt a RESET lapka alatt egy egyszerű kis érintkezőt találunk. Ennek két végpontjához hozzáforrasztjuk a vezetékeket, majd kivezetjük a gép burkolatán kívül elhelyezett pillanatkapcsolóhoz.

Gépemen például a kapcsolót a botkormány-csatlakozó „kupakjához" ragasztottam, és a kéteres ragasztott vezetéket a NYÁK széléhez kis szigetelőszalag-csíkokkal erősítettem fel. Szerencsére a kapcsoló nincs útban, s a kupakot is nehezebb elveszteni.

A megoldás azóta is beválik, és a kapcsoló egyszeri megnyomása után nem kell azon izgulni, hogy vajon elszáll-e a program. Természetesen kétszeri gombnyomásra teljes RESET hajtódik végre, és a régi gomb is ugyanúgy működőképes marad.

Nem árt arról sem meggyőződni még beszerelés előtt, hogy a pillanatkapcsoló tényleg csak a gombnyomás idején zárja az áramkört és nem fordítva.

Természetesen az átalakítással érdemes megvárni, amíg a garanciaidő lejár, és csak utána szabad leszedni a visszpecsétet.

VÁMOS SÁNDOR

Korszakváltás, *de hogyan...?!*

Bizonyára az olvasók közül is néhányan hallottak az 1988. augusztus 19-i *Korszakváltó* vetélkedőről. Ez olyan történelmi játék volt, amelynek keretében az ország különböző pontjain lévő csapatok versenyeztek egymással. A kapcsolatot rádióval és egy számítógépes rendszerrel tartották, vagyis ... Minden nagyon szép és jó lett volna, ha ...

A valóság azonban egy kicsit mást hozott, mint amit a rendezők vártak. A rádiós kapcsolat még csak létezett, de a számitógép körül nem jött minden össze. Ezt egy ún. MUPID rendszerrel szerették volna megvalósitani, s így telefonvonalakkal összekötni a helyszineket. Az elgondolás nagyszerű volt, de sajnos figyelmen kívül hagyta a mai telefonhelyzet problémáit. Ezek után úgy gondolom, mindenki sejti már a fejleményeket is. Nos, mi Siófokon voltunk, s az alábbiak az ott történteknek a leírása. Mellesleg megjegyzendő, hogy az eset egyáltalán nem kirívó, sőt, nagyon is általános.

1988. augusztus 18. Délelőtt érkeztünk a helyszínre, s így bőven volt időnk a gépeket a helyükre tennünk. Betöltöttük a programot a gépbe, majd vártuk a 13.00—13.30-ig tervezett főpróbát. A megadott időben megszólal a telefon. Első kísérletre bejutunk a rendszerbe. Örömünk azonban korai volt. Tíz másodperc elteltével lefagy a rendszer. Újabb hívás — várjunk —, vártunk fél négyig. Majd a telefonközpontos bejelentette, hogy sajnos már nem tud kapcsolni, visszaadja a vonalat. Saját hívásra viszont sikeredett a rendszerbe kerülés. Üzenet mindenkinek, hogy vagytok, milyen utatok volt stb ... Majd egy körlevél mindenkinek: ki hisz abban, hogy holnap sikeres lesz a számítógépes kapcsolat? A válaszok többnyire a MI NEM üzenetet tartalmazták. Félóra múlva a központ kitörölte a körlevelet, majd nyomatékosan megkért mindenkit, hogy ne használjuk ezt az információküldési módot, a gondokat holnapra megoldják.

Augusztus 19. Az előkészületek ugyan nem voltak zökkenőmentesek — a magnót hallani lehetett a telefonban, s néha a rádió is foglaltat jelzett —, de reggel kilenckor már várhattuk a központ hívását. Hiába. Már elmúlt tíz óra, mikor végre bejutottunk a rendszerbe. Tíz percre. Az első feladatot ugyan utólag mondták el telefonon, de a megoldást elküldtük. Többet azonban nem tudtunk tenni, ugyanis a nap folyamán mintegy háromnegyed órát használhattuk a gépet, 2-3 perces intervallumokban, ami épphogy elég volt a bejelentkezésre, de a válaszküldésre már kevésbé. Így aztán minden létező lehetőséget megragadtunk az eredmények Budapestre küldésére (telefonon, telex stb ...). Déltájban a játékvezetőség úgy döntött, hogy a tizenöt helyszínből tíznél a számítógépes vonalat megszünteti, s egyszerű telefonvonalra kapcsolja át. Ezt velünk ugyan közölték, de mivel a Petőfi adó időnként élőadásban kapcsolódott be a játékba, a nagyközönség is igényelte volna a teljes informáltságot. Végül az unszolásnak engedve kiderült a probléma technikai oldala. Hasonló volt a helyzet a televízióval is. Néhány helyszínen felvétel készült a másnapi adáshoz, s függetlenül attól, hogy az összeköttetés már rég nem létezett, a műsorban azért bemutatták a számítógépes munkát is. Ez persze a kívülállóknak aztán fel sem tűnt. Mindezek ellenére sikeresen lezajlottak a délutáni fordulók is. A játék kinek jobban, kinek kevésbé, de szerencsésen befejeződött. Igaz, az egyik feladat szerint tüzet is kellett rakni, annak ellenére, hogy ezt több helyszínen tiltották, de baj nem lett ebből.

A későbbiek folyamán beszélgettünk olyanokkal is, akiknek minden rendben ment. Ők meg voltak ugyan elégedve a dologgal, de azt nehezményezték, hogy maximum hatvanöt percig tudtak rendszerben maradni, utána a központ automatikusan kiiktatta őket.

Szomorú tapasztalat, hogy ilyen nagy rendezvény megtartásához alkalmatlan a mai telefonhálózat. Persze ez nem csupán a posta bűne, állítólag a számítóközpont sem volt tökéletes a verseny idején. Volt, aki utánajárt az ügynek a a VEIKI-nél. Miért nem tudott kézi hívással belépni a rendszerbe, holott minden feltétel adott volt? A nyomozás kemény próbának bizonyult. Először is nem tudták biztosan megmondani, hogy melyik osztályt kell az ügyben keresni, s ha akadt is valaki, akinek némi köze volt a dologhoz, csak hosszú idő után mutatta jelét, hogy sejti miről van szó. Mindez azonban csak növelte az utórezgések feszültségét. Mert tudtak róla, hogy valamilyen gépeik részt vettek ugyan valamilyen vetélkedőn, de semmi másról, ugyanis az esetleges felelősséget elhárították maguktól. Ezek után felmerül a kérdés: lesz-e jövőre Korszakváltó?

Bártfai Barnabás

```
   1 CLEAR 59999
   5 GO SUB 200
  10 LET A=61090
  20 POKE A,128
  30 LET A=A+1
  40 FOR R=1 TO 30
  45 CLS
  50 READ A$
  55 PRINT AT 0,0;a$
  60 FOR B=A TO A+LEN A$-1
  70 POKE B,CODE A$(B-A+1)
  80 NEXT B
  90 POKE B,141
 100 LET B=B+1
 110 LET A=B
 120 NEXT R
 130 POKE 59996,193: POKE 59997,237
 140 POKE 23610,28
 145 GO TO 9999
 200 FOR a=15616 TO 16383: POKE a+44384,
PEEK a: NEXT a
 210 FOR a=60768 TO 61089: READ b: POKE
a,b: NEXT a
 290 LET w=59744
 300 FOR a=1 TO 16: READ x: FOR b=0 TO 7
: READ y: POKE w+(x*8)+b,y: NEXT b: NEXT
 a
 400 POKE 23606,96: POKE 23607,233
 500 RETURN
8500 DATA 253,54,49,2,205,149,23,205,176
,22,62,0,205,1,22,205,44,15,205,23,27,25
3,203,0,126,32,18,253,203,48,102,40,64,4
2,89,92,205,167,17,253,54,0,255,24,221,4
2,89,92,34,93,92,205,251,25,120,177,194,
80,238,223,254,13,40,192,253,203,48,70,1
96,175,13,205,110,13,62,25,253,150,79,50
,140,92,253,203,1,254,253,54,0,255,253,5
4,10,1,205,138,27,118,253,203,1,174,253,
203,48,78,196,205,14,58,58,92,60,245,33,
0,0,253,116,55,253,116,38,34,11,92,33,1,
0,34,22,92,205,176,22,253,203,55,174,205
,110,13,253,203,2,238,241,71,254,10,56,2
,198,7,205,239,21,62,32,215,120,17,162,2
38,205,10,12,237,75,69,92,205,27,26,62,5
8,215,253,78,13,6,0,205,27,26,205,151,16
,58,58,92,60,40,27,254,9,40,4,254,21,32,
3,253,52,13,1,3,0
8510 DATA 17,112,92,33,68,92,203,126,40,
1,9,237,184,253,54,10,255,253,203,1,158,
195,106,237,62,16,1,0,0,195,209,237,237,
67,73,92,42,93,92,235,33,72,238,229,42,9
7,92,55,237,82,229,96,105,205,110,25,32,
6,205,184,25,205,232,25,193,121,61,176,4
0,40,197,3,3,3,3,43,237,91,83,92,213,205
,85,22,225,34,83,92,193,197,19,42,97,92,
43,43,237,184,42,73,92,235,193,112,43,11
3,43,115,43,114,241,195,96,237
9000 DATA 33,8,62,8,8,8,62,0
9010 DATA 37,8,60,74,66,126,66,66,0
9020 DATA 38,16,16,56,4,60,68,60,0
9030 DATA 39,20,86,66,66,66,66,60,0
9040 DATA 63,16,16,56,68,120,64,60,0
9050 DATA 64,16,16,0,48,16,16,56,0
9060 DATA 91,20,60,86,66,66,66,60,0
9070 DATA 92,20,60,66,66,66,66,60,0
9080 DATA 93,40,40,56,68,68,68,56,0
9090 DATA 95,20,66,66,66,66,66,60,0
9100 DATA 96,8,126,72,124,64,64,126,0
9110 DATA 123,40,0,56,68,68,68,56,0
9120 DATA 124,16,16,56,68,68,68,56,0
9130 DATA 125,40,40,68,68,68,68,56,0
9140 DATA 126,8,60,74,66,66,66,60,0
9150 DATA 127,40,0,68,68,68,68,56,0
9999 DATA "Ok?","NEXT utas@tás FOR n?lkü
l","Nincs ilyen változ!","Helytelen inde
x","Nincs hely a memóriában","Nincs hely
 a k?pernyőn","Tul nagy szám","RETURN ut
asítás GO SUB n?lkül","File v?ge","STOP
utasítás","r?nytelen argumentum","A ta
rtományon kívül esj szám","A BASIC-ben i
lyen nincs","Megszakítás-a CONT ism?tel"
,"Nincs több adat a DATA-ban","?rv?nytel
en file n?v","Nincs hely a sor számára",
"STOP az INPUT sorban","FOR utasítás NEX
T n?lkül","?rv?nytelen I/O eszk?z","?rv?
nytelen szín","Megszakítás a futás k?zbe
n","Helytelen RAMTOP ?rt?k","Már nem l?t
ez] utasítás","?rv?nytelen stream","FN u
tasítás DEF n?lkül","Param?ter hiba","Hi
ba a kazettáról bet(lt?sn?l","," ","1
988 CsRG Soft"
```

MAGYAR NYELVŰ HIBAÜZENETEK

A programmal bármilyen hosszú üzenetet definiálhatunk ZX-Spectrumon, csak ügyeljünk arra, hogy a hibaüzenet ne tartalmazzon 128-nál nagyobb ASCII kódú karaktert, mivel ez az üzenet végét jelzi. Így az üzenet valódi, utolsó kódját úgy kapjuk meg, hogy kivonunk belőle 128-at.

A program tartalmaz még egy ékezetes karakterkészletet is:

ASCII KÓD	KARAKTER	ASCII KÓD	KARAKTER
37	Á	91	Ő
38	á	93	ő
96	É	92	Ö
63	é	123	ö
33	Í	95	Ű
64	í	127	ü
126	Ó	125	Ü
124	ó	39	ű

A program NEW hatására kikapcsol. Beindítani a következőképpen lehet:

POKE 59996,193:POKE 59997,237

POKE 23606,96:POKE 23607,233

A listában található csillagok helyett egy '(c)' karaktert kell begépelni.

Csoma Roland

MI MIBŐL SZÁRMAZIK,

avagy a nyelvcsaládfáról...

Gyakran hallhatunk számítógépes berkekben dúló nagy vitákról, amiket programnyelvek művelői folytatnak, s természetesen ki-ki a saját maga által használt nyelvre esküszik. Az persze nem vitás, hogy minden nyelvnek van előnyös és hátrányos oldala is, de most nem ez a lényeg. *Írásommal nem részletezni akarom a nyelveket, hanem feladatorientáltan áttekinteni.* Tehát ne nyelvismertetést várjon az olvasó!

Minden gép csak a számára értelmezhető bifórmációkat érti meg. Gondoljuk el, hogy ha BASIC-ben leírjuk a következő utasássort: FORi=1TO10:PRINTi:NEXTi, az hány rendszerrutin hívását idézi elő fordításnál, illetve ha interpreter van, akkor futásnál. Az interpreter=értelmező; a mikrogépek általában interpreterrel vannak ellátva (lásd BASIC). A compiler=fordító; ilyenkor a tárgyprogram csak egyszer fordítódik le, és ezek után x-szer futtatható.

No de visszatérve a nyelvekhez, alapul a gépi kódot kell venni, ahova az assembler is sorolható, ami nem más, mint a gépi kód utasításainak karakteres szintaktikában való leírása.

A rendszerprogramozók — már akik nyelveket fejlesztettek — több szempont szerint alkottak, pontosan három szerint, amelyek végül matematikai, adatfeldolgozási és általános feladatokra orientálódtak.

Később a matematikai ág egybeolvadt az adatfeldolgozóval. Ilyen matematikai, de az adatfeldolgozó nyelvek csoportjában található az ALGOL60—68 és a Modula—2. Ezekben a nyelvekben olyan utasítások vannak, amelyek nagymértékben építenek a gép aritmetikai műveleteire, és logikai megoldókészségére is, ahol is nagyon fontos a műveleti sebesség.

Ebben az adatfeldolgozó ágban találhatjuk azokat a nyelveket is, amelyek igen magas szintűek, de mind az adatfeldolgozást, mind a matematikai műveleteket használják, és mégsem általános nyelvek. Ezek a PL/I, a Pascal és az Ada. Persze az igazán adatcentrikus nyelv a COBOL és az OCCAM.

Mint minden nagy, illetve óriási adattömeghez, szükséges a jól szervezett fájlkezelés is, tehát ezek a nyelvek ezt messzemenően támogatják.

A másik nagy csoport, vagyis ág az általános célú nyelveké. Ennél a fejlesztők arra törekedtek, hogy az adott hardverbázis összes lehetőségét elérhesse a felhasználó, ezért viszont fel kellett áldozniuk a részletességet — az assemblerrel ellentétben, ahol minden elérhető —, de ezért nagyon egyszerűen kezelhető, magas szintű nyelvet kap a felhasználó. Erre tipikus példa a BASIC, amely rendelkezik matematikai függvényekkel, fájlkezelő utasításokkal, adatfeldolgozó parancsokkal, grafikai lehetőségekkel, hanggenerálással stb.-vel.

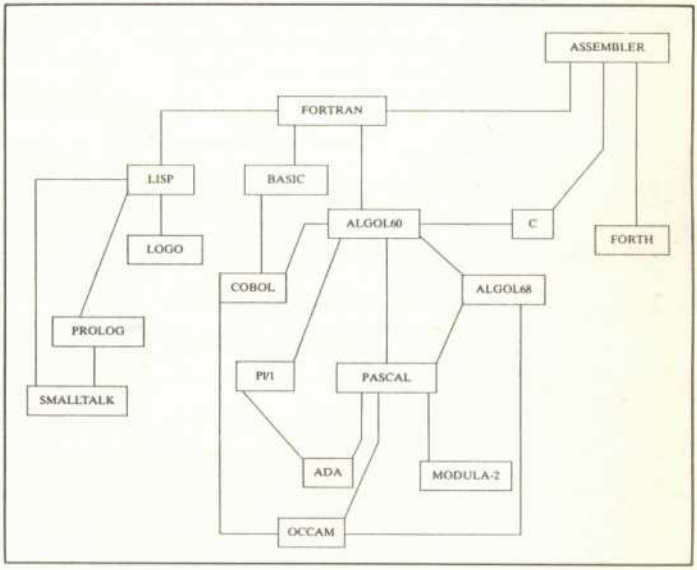
Igazán azonban egyiket sem lehet használni nagyobb feladatnál! Sok programozó nem szereti a BASIC-nyelvet az említett hibák miatt, főleg, hogy általában PC-n, illetve nagygépes rendszerekben több különböző nyelven írt fájlok összefűzhetők egy értelmes programfájllá. Egyébként én is ezzel az elvvel értek egyet, annak ellenére, hogy szerintem a BASIC nagyon jó arra, hogy megismerkedve vele, a kezdők látják, mi mindent lehet elérni a számítástechnikával. De a BASIC-tudással nem szabad megelégedni!

Az általános nyelvek új irányzatot követnek a nyolcvanas évek eleje óta, amikortól kezdve ugyanis a grafika kapott nagyobb teret, és ezért egyre inkább követelménnyé vált a látvány a tudás mellett, de nem ennek terhére. Ezért a programnyelvek is kezdték kihasználni ezt a lehetőséget is, mint a LOGO és a G-Pascal.

A jövő mindenesetre a grafikáé, tehát az biztos, hogy az ikonvezérlés nem újdonság a PC-k világában, ahol már a DOS is így működik, például az Apple PC-kben.

BARTOS GYULA

(A CHIP Professional ábrája alapján)



Listánkat felhasználói, illetve játékprogramokból állítjuk össze. A legjobbakat, legérdekesebbeket a beküldött javaslatok alapján rangsoroljuk. Ehhez kérjük az olvasók közreműködését. C64-re, ZX-Spectrumra. Enterprise-ra, ATARI-ra és IBM-re készült programrangsorokat várunk havonta.

Címünk: Mikroszámítógép Magazin Szerkesztősége 1371 Bp., Pf. 433.

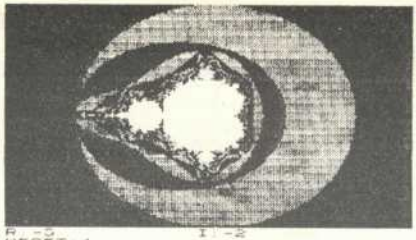
Diákszerkesztőség

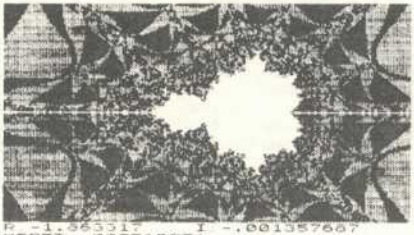
TOP - lista

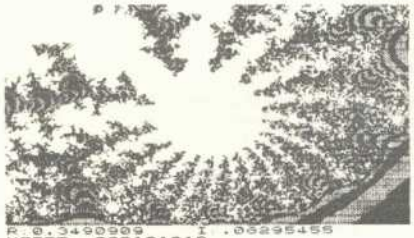
BBS

Felhasználói		Játék	
1. News room	C64	1. Defender OTC.	C64 Amiga
2. GEOS 1.32	C64	2. Pentis	IBM
3. Newsmaster	IBM	3. Test Drive	C64
4. IS DOS	Enterprise	4. Flight sym. II.	C64
5. Printmaster	C64 IBM	5. The Last Ninja	C64 Spect.
6. Windows	IBM Ent.	6. Skyfox II.	C64 Spect.
7. 128 Starpainter	C128	7. Out run	C64
8. Rockmonitor III	C64	8. Impossible m. II.	C64 Spect.
9. Printfox	C64	9. The Guild of t.	C64
10. Game maker	C64	10. Blood and guts	C64

```
 10 REM        MANDELBROT
 20 REM
 30 REM      TURBO-SOFT 1988
 40 REM
 50 DIM S$(15): LET W=256/176
 60 FOR I=0 TO 13: READ A: POKE 23296+I,A
: NEXT I
 70 POKE 23658,8: BORDER 7
 80 INPUT "LOAD VAGY INPUT "; LINE T$
 90 IF T$="L" THEN  GO SUB 440: GO TO 230
100 INPUT "R:";A: INPUT "I:";B: INPUT "M:
";O
110 LET SW=0: LET P=O/176: BORDER 7: CLS
120 PRINT #1;"R:";A;TAB 15;"I:";B: PRINT
#1;"MERET:";O;
130 FOR N=0 TO 175: LET CI=B+N*P
140 FOR M=0 TO 255: LET CR=A+M*P
150 LET X=0: LET R=0: LET I=0
160 LET R1=R*R-I*I+CR: LET I=2*R*I+CI: LE
T R=R1
170 LET Z=SQR (R*R+I*I): LET X=X+1
180 IF NOT ((X=100) OR (Z>2)) THEN  GO TO
160
190 IF X=100 THEN  GO TO 220
200 LET S=((X/2)-INT (X/2))*2
210 IF S OR SW THEN  PLOT M,N
220 LET SW=NOT SW: NEXT M: LET SW=NOT SW:
NEXT N
230 LET CA=0: LET X1=128: LET Y1=88: LET
Y2=0
240 GO SUB 500
250 LET A$=INKEY$
260 GO SUB 500
270 IF A$="" THEN  GO TO 240
280 IF A$="S" THEN  POKE 23307,194: POKE
23308,4: RANDOMIZE USR 23296
290 IF A$="L" THEN  GO SUB 440
300 IF A$="N" THEN  LET CA=0: BORDER 7
310 IF A$="6" THEN  LET X1=X1-1: LET Y1=Y
1-CA/W
320 IF A$="7" THEN  LET X1=X1+1: LET Y1=Y
1+CA/W
330 IF A$="9" THEN  LET Y1=Y1+1: LET X1=X
1+W*CA
340 IF A$="8" THEN  LET Y1=Y1-1: LET X1=X
1-W*CA
350 IF A$="0" AND CA THEN  LET A=A1: LET
B=B1: LET O=ABS ((Y1-Y2+1)*P): GO TO 110
360 IF A$="0" AND NOT CA THEN  LET CA=1:
BORDER 0: BEEP 1,1: LET A1=A+P*X1: LET B1=
B+P*Y1: LET Y2=Y1: LET X2=X1
370 IF A$=" " THEN  GO TO 70
380 IF X1<0 THEN  LET X1=0
390 IF X1>255 THEN  LET X1=255: IF CA THE
N  LET Y1=Y2+(X1-X2)/W
400 IF Y1>175 THEN  LET Y1=175: IF CA THE
N  LET X1=X2+(Y1-Y2)*W
410 IF Y1<0 THEN  LET Y1=0
420 IF CA*(Y1<Y2) THEN  LET Y1=Y2: LET X1
=X2
430 GO TO 240
440 LET S$="": POKE 23307,86: POKE 23308,
5: RANDOMIZE USR 23296
450 FOR I=2 TO 14: LET S$(I-1)=SCREEN$ (2
2,I): NEXT I: LET A=VAL S$
460 LET S$="": FOR I=17 TO 31: LET S$(I-1
6)=SCREEN$ (22,I): NEXT I: LET B=VAL S$
470 LET S$="": FOR I=6 TO 20: LET S$(I-5)
=SCREEN$ (23,I): NEXT I: LET O=VAL S$
480 LET P=O/176: LET CA=0: BORDER 7
490 RETURN
500 OVER 1: PLOT X1,Y1
510 IF CA THEN  DRAW X2-X1,0: DRAW 0,Y2-Y
1: DRAW X1-X2,0: DRAW 0,Y1-Y2
520 OVER 0: RETURN
530 DATA 221,33,0,64,17,0,27,62,255,55,20
5,194,4,201
```







ILYEN VOLT, ILYEN LETT

MANDELBROT

Ezzel a programmal a Mandelbrot-halmaz különböző részeit nagyíthatjuk ki a Spectrum képernyőjére. A halmaz egy területének megadásához három adatra van szükség: R a nagyítandó téglalap bal alsó sarkának X koordinátája, I az Y koordinátája, a MERET pedig a téglalap Y irányú oldalhosszúsága. A terület számításának alapadatai egy-egy ábra alatt láthatók.

Az alaphalmaz az R= −3, I= −2, MERET=4 képben helyezkedik el. A fehér terület nem része a halmaznak. Az érdekes képek mindig a halmaz szélén helyezkednek el.

A program indítása után döntenünk kell, hogy egy korábban elkészített képet nagyítunk tovább, vagy koordinátáik alapján készítünk egyet. Ha kazettáról töltünk be képet, akkor a program az adatait automatikusan megállapítja. Amikor a gépben egy kész kép van, akkor ezt a kép közepén vibráló pont jelzi. Ilyenkor kimenthetjük a képet kazettára S-sel, vagy betölthetünk egyet L-lel. Tovább nagyíthatunk is: a vibráló pontot Interface II botkormánnyal mozgassuk a nagyítandó terület bal alsó sarkába (6,7: bal, jobb; 8,9: le, fel) és nyomjuk meg a tűzgombot (0). Ekkor a border feketére vált, és a nagyítandó terület jobb felső sarkát tudjuk mozgatni. Ha a bal alsó sarok nem a megkívánt helyen van, akkor az N billentyű megnyomása után ismét a bal alsó sarok helyét lehet mozgatni. A jobb felső sarok beállítása után a tűzgomb hatására elkezdődik a nagyítás. Ez a művelet — különösen akkor, ha sok a fehér terület — elég időigényes, ezért érdemes a programot lefordíttatni egy olyan fordítóval, amelyik kezeli a törtszámokat (ilyen például a TOBOS FP). Lefordítva egy kép egy-négy óra alatt készül el.

NAGY ZOLTÁN

Egy sarokkal olcsóbb!!

Mikroszámítógép Magazin 1988. december

ENTERPRISE® COMPUTERS GMBH

szelvény

Az ENTERPRISE COMPUTERS GMBH cég felajánlotta szerkesztőségünknek, hogy részt vesz árengedményes akciónkban. Aki Münchenben jár — ami a világútlevél korában már nem csupán vágyálom — és leadja a magazinból kivágott sarokszelvényt, az az alább felsorolt ENTERPRISE gépek, tartozékok, könyvek árából 15% engedményt kap. Egy sarokszelvény — az eddigieknek megfelelően — csak egy-egy tétel engedményére jogosít. Tehát ha valaki egy csomag kazettát és esetleg egy könyvet is vásárol, annak két szelvényt kell leadnia.

A CÉG AJÁNLATA:

ENTERPRISE 128K (német BASIC cartridge)	*298 DM*
ENTERPRISE 64K (angol BASIC cartridge)	*198 DM*
EXDOS floppyvezérlő+ kézikönyv	*148 DM*
FLOPPY EGYSÉG (5,25 inch, egyoldalas, 40 sávos, 180 Kbájtra formattál)	*198 DM*
FLOPPY EGYSÉG (5,25 inch) + vezérlőegység	*298 DM*
„EP80+ " típusú nyomtató (100 kar/s)+ illesztőkábel	*348 DM*
SPEAK—EASY beszédszintetizátor	*69 DM*
Mindegyik programkazetta	*15 DM*
Programkazetta-csomag (3 db)	*40 DM*
Programkazetta-csomag (10 db)	*100 DM*
BASIC oktatókönyv	*19 DM*
EXDOS kézikönyv	*19 DM*
15 DOS kézikönyv+ floppy	*29 DM*
EXOS (műszaki kézikönyv)	*39 DM*
ENTERPRISE 128K kézikönyv (265 oldalas, német nyelvű)	*10 DM*
Nyomtatókábel (CENTRONICS, 8 bites, párhuzamos)	*24 DM*
Botkormány-kábel	*24 DM*
Monitorkábel (SCART, CYNCH csatlakozóval vagy anélkül)	*24 DM*
RS232C kábel soros nyomtatókhoz	*24 DM*

A müncheni cím:
ENTERPRISE COMPUTERS GMBH
8000 MÜNCHEN 2.
SONNENSTRASSE 3.
NSZK

BASIC ÉS GÉPI KÓD

Legutóbb a bináris számokat BCD-re alakító rutinnak egy olyan változatát ismertettem, amely a szám elején álló értéktelen nullákat elhagyja, majd néhány, a téma további alkalmazására vonatkozó ötlet felvetésével befejeztem a BCD-aritmetika bemutatását.

Most az augusztusi számban megjelent írásommal kapcsolatos olvasói észrevételekkel és azok tanulságaival foglalkozom.

Három levelet kaptam. Néhányan szóban fejtették ki véleményüket munkahelyemen, a Magiszter Könyvesbolt számítástechnikai részlegén.

Abban mindenki egyetértett, hogy a programlistát és a hozzá tartozó magyarázatot nem szabad elszakítani egymástól. A bemutatott listákkal kapcsolatban megoszlanak a vélemények.

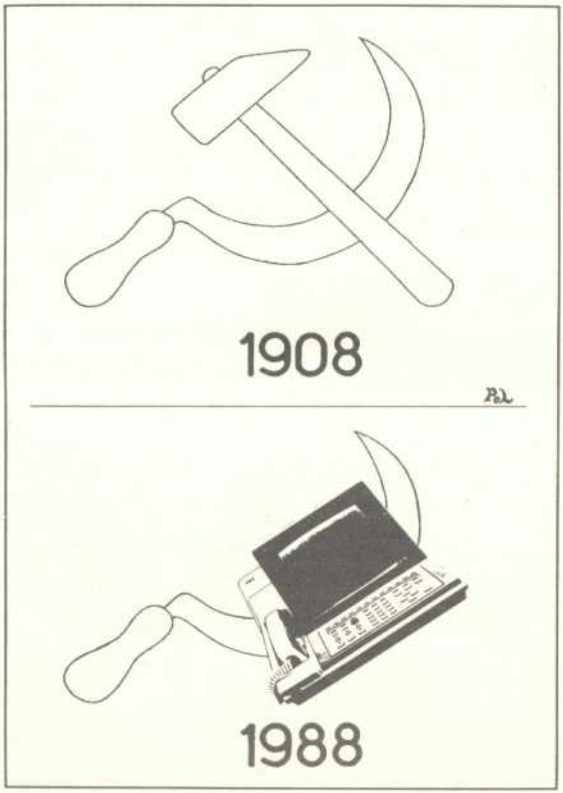
Szabó Péter Pál levelében az augusztusi számban közölt 1. és 2. programlista helyett azok kombinációját ajánlja, hasonlót a HELP-PLUS, illetve a PROFI-ASS 64 kimeneti listájához. Ifjabb Fekete László a disassemblált alakra szavaz, az assembly és szimulátorlista közlését felesleges helypazarlásnak tartja. Scherr Zoltán szerint a szimulátor listája is hasznos lehet néha.

A véleményeket egybevetve, Szabó Péter Pál javaslatát tartom a legcélszerűbbnek, bár a megvalósítással valószínűleg gondjaim lesznek.

Míg Szabó Péter Pál csak a technikai jellegű kérdéseimhez szól hozzá, a másik két levélíró a cikksorozat folytatása mellett foglalt állást.

Ifjabb Fekete László hasznos tanácsokat ad, ezenkívül rengeteg témát sorol fel, amelyekről szívesen olvasna. Remélem, ötleteit sikerül megvalósítanom.

Scherr Zoltánnak VC20-as gépe van. „... feltétlenül legyen figyelemmel a VC-20-as sajátosságaira a továbbiakban is, mint eddig" — írja többek nevében is. Levelének más részében a következőket olvasom: „A korábban megjelent információk ismételgetését felesleges nek tartom. Kivételes esetekben is elég a megjelenés helyére utalni." Könyvesbolti beszélgetőtársaim többsége éppen ellenkező véleményen volt, és véleményük alapján tudatosult bennem, hogy mit hibáztam el, és miért nem érdemes ezt a cikksorozatot a jelenlegi formájában folytatnom.



A „Gép is ember" c. karikatúrapályázat harmadik helyezett alkotása, LOUIS POSTRUZIN munkája.

Eredeti elképzelésem az volt, hogy az alapoknál kezdve úgy építem fel cikkeimet, hogy minden új információ a már korábban közöltekre támaszkodjon. Csupán arról feledkeztem meg, hogy ez a módszer egy könyvben jó lehet, de egy folyóiratban megjelenő cikksorozat esetén teljesen elhibázott. Új olvasók kapcsolódnak be, akik, nem ismervén az előzményeket, értetlenül fogadják az újabb ismereteket is. Még annak is nehéz, aki a lap korábbi számainak birtokában próbál újonnan bekapcsolódni a sorozat olvasásába, hát még annak, akinek a korábban megjelent példányok hozzáférhetetlenek.

A másik hiba a munkamódszeremben rejlett. A sorozatot folytatásosként írtam, legtöbbnyire a szorongató határidő árnyékában. Néha ez volt az oka a programlista és a magyarázat különválásának is. A „határidőre kész kell lenni, mert egy folytatás sem maradhat ki" hangulata olykor kapkodáshoz vezetett. Előfordult, hogy „raktárból" vettem elő egy, a korábbiakkal össze nem függő témát, mert a szükséges program nem akart „összeállni".

A harmincegyedik folytatással befejezem az összefüggő *BASIC és gépi kód* című sorozatot. A cím néha megjelenik a lap hasábjain, de ritkábban, és csak akkor, ha valóban van új mondanivalóm. Újabb cikkeket tervezek a BASIC és gépi kódú programozás témaköréből (most már külön-külön), de ezekkel csak akkor jelentkezem, ha „teljesen kész van az anyag".

Köszönöm az olvasók türelmét, és külön köszönöm a levélíróknak, mert nem sajnálták a fáradtságot, hogy segítsenek.

BARNA LÁSZLÓ

A PROGRAMOZÁS ALAPKÉRDÉSEI IV.

Hibajelzések és üzenetek

Ötrészes sorozatunkban mindarról szót ejtünk, amit olvasóinknak érdemes átgondolniuk, ha programozásra adják a fejüket, vagy ha valamely programról véleményt kívánnak alkotni. Ezek az írások a szerzőnek lapunkban közölt korábbi fejtegetéseivel együtt (utalunk „Az adattípus fejlődése" és a „Függvények és utasítások" című sorozatokra, lásd 1987/8. és 9., valamint 1987/12., 1988/1. és 2. számainkat), teljeskörűen és közérthetően elemzik e sorozat címében jelzett (problematikát) témát.

Tegyük fel, hogy van egy teljesen hibátlannak vélt programunk. Vajon száz futásból hány futása lesz hibátlan? Programja válogatja. Van, amelyiknek kilencven, van, amelyiknek húsz.

Képzeljük el a legideálisabb esetet, egy bemenő adatot nem igénylő, egyszerű kis programocskát. Például azt, amelyik betölti a nyomtatóba a magyar karakterkészletet. Ezt a programot gondosan megírták, belőtték, és naponta legalább egyszer gond nélkül használják. Ha elindítjuk, biztosan jól működik, kivéve, ha nincs bekapcsolva a nyomtató, vagy off line-ba van kapcsolva, vagy kilazult a csatlakozó.

Kivételes esetben a következők fordulhatnak elő: a program lefut, mintha mi sem történt volna; a program „abortál" valamilyen nehezen érthető üzenet kíséretében (például „nyugtázatlan üzenet" vagy „nem jött meg a 'done' bit" vagy „ERROR 2749" stb.); a program szó nélkül leragad és addig vár, amíg rá nem jövünk, mi a hiba; végül jöhet valami értelmes üzenet is, például mindössze ennyi: „nézd meg, mi van a nyomtatóval".

Vegyünk egy példát a másik végletből is, mondjuk egy fordítóprogramot. Általános tapasztalat, hogy legalább tízszer el kell indítani a fordítóprogramot, amíg a felhasználó programja először futóképes lesz, és utána minden javításnál kétszer, háromszor.

A fentiekből egyrészt az következik, hogy nincs olyan program, amelyben a hibajelzésnek nincs szerepe, másrészt pedig az, hogy különböző programoknál különböző súlya van a dolognak.

A MINIMÁLIS TENNIVALÓ

A hibakezelésnek szükségszerű feltétele, hogy a hibát észrevegyük. Rengeteg időt és energiát takaríthatunk meg azzal, hogy akkor vesszük észre, amikor keletkezik, és nem akkor, amikor a következményei mutatkoznak. Egy olyan hiba megkeresése a következmények alapján, amelyen átsiklottunk, rendkívül nehéz és bonyolult feladat még a program írójának is, nem is szólva arról, amikor más programjában kell a hibát megtalálni.

Gondoljunk arra, hogy egy hibásan kinyomtatott lap legalább 1 forintba kerül, egy óra gépidő 1000 forintokban mérhető, nem is beszélve saját idegeinkről, vagy — legrosszabb esetben — a téves pályára állt rakétáról.

Jegyezzük meg: most nem a program hibáiról van szó, hanem azokról a hibákról, amelyek azért lépnek fel, mert a program kapcsolatba kerül a külvilággal. Világos, hogy ezt a kapcsolatot ellenőrizni kell. A kapcsolat néha hardver eszközökkel jön létre, de ez a ritkább eset. Akik számítógépes adatgyűjtéssel, folyamatirányítással és hasonló dolgokkal foglalkoznak, nagyon jól tudják, hogy milyen fontos a bejövő adatok hitelességének, illetve hihetőségének ellenőrzése.

A közönséges programok többsége a kapcsolatot a külvilággal a bemenő adatokon keresztül érzékeli. A bemenetet is két részre lehet osztani: az előre elkészített adatokra és a felhasználóval való közvetlen kapcsolatra. A bemenő adatokat ellenőrizni kell — bárhonnan is jönnek —, de természetesen nem reagálhatunk egyformán egy előre fájlba készített adatra és arra, amikor egy élő emberrel folytatunk dialógust, és így nyerünk adatokat. Vannak olyan rendszerek és programok, amelyek egyenlőségjelet akarnak tenni a két kapcsolat közé, ezáltal valamilyen univerzalitás látszatát igyekszenek kelteni, de az eredmény igen siralmas.

AZ ELŐRE ELKÉSZÍTETT ADATOK ELLENŐRZÉSE

Az előre elkészített adatok ellenőrzésének legfontosabb eszköze a redundancia. Ez azt jelenti, hogy a programnak nemcsak a működéséhez szükséges adatokat adjuk meg, hanem további adatokat is, amelyek pusztán az előzőek ellenőrzésére szolgálnak.

Mindig ideges leszek, ha olyan BASIC programot látok, amely több száz DATA-val megadott adatot tartalmaz, és ezt a program négy-öt különböző helyen, minden ellenőrzés nélkül olvassa be. Sajnos mind a magyar, mind a külföldi folyóiratok, sőt a könyvek is tele vannak ilyen összebarkácsolt programokkal. A professzionálisan elkészített programok a következőképpen csinálják ezt. Először is elhatárolják egymástól a program különböző részeihez tartozó adatokat, a beolvasás előtt ráállnak a programrészhez tartozó első adatra, és a beolvasás végén ellenőrzik, hogy az utolsó adatot olvasták-e be. Ha az adatmennyiség tetemes, közbülső ellenőrző pontokat is beiktathatnak.

A beolvasandó adatok végét, illetve a közbülső ellenőrző pontokat úgy találjuk meg, hogy az adatok közé „mérföldköveket" helyezünk. Például ha a bemenő adatok csupa számból állnak, akkor mérföldkőnek az egyes sorozatok végére egy-egy sztringet illesztünk (például "*** 1 ******", "*** 2 ***" stb.).

Ha valahol kimaradt egy adat vagy egy vesszű, a beolvasás típushiba miatt megakad. Ha viszont kétszer írtunk be egy adatot, akkor az utolsó beolvasás nem az ellenőrző sztringet találja meg, és a program maga fog hibát jelezni. Mindkét esetben világos az utolsó ellenőrző sztring tartalmából, hogy meddig volt jó a bemenő adatok sora és hol kell keresni a hibát.

Ez természetesen igen primitív ellenőrzés, hiszen csak az adatok számát ellenőrzi; szokásos még a sorozat utolsó tagjául egy kontrollszummát adni, mert így meg lehet nézni, hogy az utolsó adat az előzőek összege-e. Ez sem érzékeny két karakter felcserélésére, de ha nagyon fontos a biztonság, valami bonyolultabb összefüggést is képezhetünk. Az ilyen megoldást viszont csak akkor szoktuk használni, amikor az adat lényegében a program szerves része és csak ritkán változik, mert az adatok ilyen bonyolult előkészítése nem kedvelt feladat. Tehát olyankor élünk az efféle megoldással, amikor a programhoz táblázatokat, szótárakat és egyéb ilyesmit olvasunk be, és bizonyosak akarunk lenni afelől, hogy az adatok nem sérültek meg, illetve, hogy jól olvassuk be őket.

A „mérföldkövek" beiktatása a nagy homogén adatmezőkbe az emberi léptékek szem előtt tartása miatt is fontos: a kiugró, eltérő formájú adat azoknak is kapaszkodó, akik előkészítik ezt a homogén, jellegtelen adattömeget.

Ha a bemenő adatoknak eleve van valamilyen struktúrájuk, természetesen nemigen van szükség ilyen mérföldkövekre: viszonylag hamar kiderül, hogy nem odaillő adat következik a sorban.

Az adatok sorrendjén és formáján kívül a program ellenőrizheti a bejövő adatok értékét is. Ezt néha fontosnak tartjuk, néha nem. Meglehetősen nagy munka ez, és a program futását is nagyon lassítja. Az Ada nyelv lehetőséget nyújt arra, hogy egy változó tipusának alsó és felső határt írjunk elő: amikor ilyen változókba írunk bele, a program automatikusan ellenőrzi, hogy a szám az adott határok közé esik-e. Ezáltal a programozási munkát megspóroltuk, de a futás lelassulása továbbra is fennáll. Az ún. részhalmaz típus segítségével ilyen ellenőrzésre a Pascal és a Modula nyelvben is lehetőség van, de csak egészeknél vagy egész jellegű értékeknél.

AZ EMBER–GÉP KAPCSOLAT ELLENŐRZÉSE

Amikor az adatok nincsenek előkészítve, hanem a program úgy működik, hogy a gép az emberrel tart közvetlen kapcsolatot, akkor a bemenetet egészen másképp kell megszervezni.

Az emberek nem szeretik, ha akár csak eggyel is többször kell leütniük a billentyűt a szükségesnél — tehát a redundanciát kerülni kell. Ezzel szemben az ember viszont elvárja, hogy állandóan tájékoztassák arról, hol tart a munka, milyen bemenő adatot kívánnak tőle. Másrészt általában nem tud gépelni, tehát meg kell adni neki a lehetőséget arra, hogy közben javítson. Majd ellenőrizni kell, mit írt be, és ha hibás, ismét lehetővé kell tenni a javítást.

Viszonylag könnyen összeállítható egy olyan eljárás, ami a következőket csinálja: kiír egy paraméterenként megadott karaktersorozatot (amely lehet üres is), engedi, hogy a végére írjunk valami folytatást és a kapott szövegbe beszúrjunk, töröljünk, és hogy végül az eredményt elküldjük a számítógépbe. Ez az eljárás igen alkalmas lesz arra, hogy az egész ember-gép kapcsolatot szervezzük vele.

Elvileg egyszer az életben elég lenne egy ilyen eljárást megírni, és akkor mindig, mindenki azt használhatná. Sajnos jelenleg ez nem járható út, mert az apró részletkérdésekben minden gép egy kicsit más konvenciókat követ, mint a másik. Az egyik gép az INSERT gomb megnyomására egy karaktert szúr be, a másik oda-vissza váltogatja a beszúró és felüliró üzemmódot. Más gombot jelöltek ki törlésre az egyik és a másik gépen, másik gombbal jelzik, hogy vége az üzenetnek, és így tovább. Még a magyar ékezetes betűk leírására is legalább tíz különböző megoldást ismerek. Van, ahol először a betűt kell leütni és azután az ékezetet, van ahol fordítva. Ezek után nem csoda, ha a felhasználó, amikor egy új, eredetileg más gépre írt programot először próbál ki, rögtön megállapítja, hogy ezt csak valami agyalágyult követhette el.

Ezért tehát, ha olyan programot írunk, amiről azt hisszük, hogy másnak is szüksége lehet rá, a bevitelt végző részt különítsük el a többi résztől amennyire csak lehet, és ezáltal adjuk meg a felhasználónak a lehetőséget arra, hogy olyanra cserélje ki, amilyen az ő ízlésének megfelel.

A HIBAKEZELÉS KÉT MÓDJA

A futás közbeni hibákat két nézőpontból közelíthetjük meg. Az első felfogás szerint a futás közben — a bevitel hibája miatt — létrejött hiba olyan halálos lövés, ami után a program egyetlen dolga, hogy elküldje a gyászjelentéseket. A másik nézet szerint a hiba támadás a program ellen, amit azonban kis szerencsével ki lehet védeni. A hiba kivédését a modern programozási nyelvek kivételkezelésnek (exception handling) nevezik.

A régebbi nyelvekben is volt lehetőség a kivédésre. A programot végrehajtó függvényeket úgy kellett megírni, hogy ne csak arra készüljenek fel, hogy a tevékenység jól végbemegy, hanem a hibákra is. Ezért a függvénynek nemcsak a szorosabban vett eredményt kell visszaadnia, hanem egy állapotjelzőt is, ami azt mutatja, hogy sikerült-e elvégezni a feladatot, és ha nem, mi volt az oka. Ezt az állapotjelzőt felhasználva értelmes üzenetekre és korrekciókra volt lehetőség. Ilyesmit leggyakrabban a periféria- és fájlkezelő függvényekbe szoktak beépíteni (például kísérlet nem létező fájl megnyitására, a periféria off line-ban van stb.).

A modern nyelvek kivételkezelése abban különbözik ettől a hagyományos megoldástól, hogy úgy tekinti a dolgot: egy függvény jó működése a normális állapot, a hibás működés pedig kivételes állapot. A program fő vonalát úgy kell megírni, hogy az csak a normális, jó működéssel foglalkozzon, amikor pedig a kivételes állapot bekövetkezik, a vezérlés automatikusan a program valamilyen speciális helyére, kivételkezelésre kerül.

Ahhoz, hogy a kivételes állapotot kezelni tudjuk, van egy sor kivétel, amelynek a programozási nyelv eleve nevet ad (túlcsordulás, rosszul működő berendezés, nem létező fájl stb.), további eseteket a programozó vezethet be, és az ezekhez tartozó kivételes állapot a program egy speciális utasításának kiadására következik be.

Ezek után egy függvény működését úgy képzelhetjük el, hogy a függvény vagy jól lefut és a jó eredményt visszaadva ott folytatja a munkát, ahol a függvényhívást kiadták, vagy hiba miatt kivételes állapotba kerül, és a vezérlés a hívó programnak arra a pontjára jut, amelyik a kivételes állapotokat kezeli. A kivételkezelő felsorolása megmondja, melyik kivételnél mit kell csinálni, és rendszerint arra is van lehetőség, hogy legutoljára bármilyen addig fel nem sorolt hibát elintézzünk.

A hiba kezelésén különböző dolgokat értenek. A MESA nyelvben arra is lehetőség van, hogy bizonyos módosítások után visszaugorjunk a függvényhívásra vagy a munkát a függvényhívás után folytassuk. Az Ada nyelvben viszont csak annyit tehetünk, hogy az adott programozási egységnek (rendszerint függvénynek vagy eljárásnak) a hiba előfordulása utáni részét egy másik utasítássorozattal helyettesítjük.

A szakemberek között vita folyik arról, hogy mi is tekintendő kivételnek és mit értsünk kezelésén. Vannak, akik egészen tágan értelmezik a kivételkezelést: egy fájl beolvasására például írtak olyan függvényt, ahol annak törzse egyetlen végtelen beolvasó ciklus, és mivel természetesen valamelyik beolvasásnál előbb-utóbb „END—OF—FILE" hiba lép fel, ekkor a kivételkezelő bezárja a fájlt és visszatér a hívó programhoz. Mások szerint ez nem becsületes program, hiszen maga idéz elő olyan hibát, amire nincs is szükség.

FARKAS ERNŐ

KONKLÚZIÓK

A futás közbeni hibák két forrásból származnak: az egyik, hogy rosszul gondoltuk át, mit kell a programnak csinálnia, a másik, hogy a környezet nem úgy viselkedik, ahogy vártuk. A két vétek majdnem azonos következményű. Nem tarthatunk egy programot jónak, ha csak ideális, steril körülmények között működik.

Az élet bonyolódik az egészen egyszerű emberi hibáktól is, és néha még a tökéletes berendezések is elromlanak. Manapság a jó program egyik fő jellemzője a robusztusság. Egy ilyen programnak meg se kottyan, ha a bemenő adatok között üres sorok vagy felesleges szóközök vannak. Érthető hibaüzeneteket küld, és az emberi kapcsolaton keresztül lehetőséget nyújt a hiba kijavítására. Amikor a program egy hibán „elszáll", nem rontja el saját adatfájljait és nem hagyja inkonzisztens állapotban a fájlkezelő rendszert.

Ezek a dolgok természetesen csak megfelelő eszközökkel, operációs rendszerekkel és programozási nyelvekkel érhetők el, de a programozónak is hozzá kell tennie a maga részét: a gondos programtervezést és kivitelezést — egyre inkább ez a világszínvonal mércéje.

A NAGYFELBONTÁSÚ KÉPERNYŐ FORGATÁSA BÁJTONKÉNT

COMMODORE 64

II.

Az előző részben a nagyfelbontású képernyő bájtonkénti forgatását ismertettük, melynek hatása azonban olyan volt, mintha csak bitenként forgattuk volna a képernyő memóriatartalmát, mivel a forgatás függőlegesen felfelé irányult, és a forgatott grafikus sorok 1 bit, azaz 1 pixel vastagságúak voltak. Most a képernyő vízszintes, X irányú, bájtonkénti forgatásáról lesz szó, amely már hatásában is mutatni fogja a grafikus kép bájtonkénti, azaz 8 pixellel balra való elmozdulását. Amit korábban mondtunk el a grafikus képernyő szerkezetéről, valamint a BASIC munkaterület 16 kbájtra átállításáról, azok itt is érvényesek.

Az *1. listán* a forgató rutin gépi kódú programja látható, amit a SYS(52480)-nal lehet elindítani. A SYS(52604) utasítás segítségével bekapcsolhatjuk a $2000-nél kezdődő grafikus képernyőt, a SYS(52621) utasítással pedig visszatérhetünk a karakter üzemmódhoz.

A program sorainak magyarázata

170—180 A két nullás lapon lévő mutató segítségével forgatjuk a képernyőt. Az AKT az éppen aktuális bájtra, a KOV pedig a jobb oldalról vele szomszédos (azaz a tőle számozásban 8 bájtnyira lévő) bájtra mutat.

200—240 A ROM-ról RAM-ra kapcsolunk át, és eközben az IRQ megszakításokat is letiltjuk, igy majd a ROM alatt elhelyezkedő Simon's BASIC képernyők forgatása is lehetségessé válik.

260—390 Kezdőértékek beállítása. Az időleges tároló, a TEMP felveszi a grafikus képernyő elejének kezdőcimét: TEMP=KEP (=$2000); a karaktersorok száma bekerül a SORSZM változóba SORSZM=24; KARBJT=7 az egy karaktersort felépítő grafikus képpontsorok számát határozza meg (0-7), mig legvégül az X változóba a képernyőn lévő oszlopok száma kerül (0-39).

410—490 A cikluson belüli értékadás: AKT=TEMP és KOV=TEMP+8.

510—530 A grafikus képsor legelső bájtját a veremtárolóba mentjük ki.

550—650 Itt jön létre a grafikus képsor balra forgatása egy bájttal. Vegyük észre, hogy ez a ciklus negyvenszer (39-től 0-ig) hajtódik végre, miközben az Y értékét 8-cal növeljük. Mivel a 40×8=320 már nem ábrázolható az Y indexregiszter egy bájtján, ezért 256-tal meg kellett növelni az AKT és a KOV mutató értékét.

670—680 A grafikus képsor veremtárolóba kimentett legelső bájtját itt irjuk vissza a veremtárolóból a képsor utolsó bájtjába.

710—750 A TEMP értékét nyolc alkalommal (0-7) megnöveljük 1 bájttal, létrehozva ezzel az új grafikus képsor kezdőértékét. Nyolc ilyen grafikus képsor alkot egy karaktersort.

780—860 A képernyőn lévő karaktersorok számának megfelelően 25 alkalommal növeljük meg a TEMP értékét 312-vel.

880—919 A RAM-ról visszakapcsolunk újra ROM-ra, és az IRQ megszakításokat is újra engedélyezzük.

930—990 Átkapcsolás a $2000-nél (8192) lévő grafikus képernyőre.

1010—1070 Visszakapcsolás a normál kijelzésre.

A *2. listán* látható a nagyfelbontású képernyőt forgató rutin BASIC betöltő programja. Ha ennek a programnak az 1050-es sorát átírjuk a következő módon:

1050 W=49152:FOR I=W TO W+157

akkor ezzel a gépi kódú program a $C000 tárterületre tölthető be. Ilyenkor a nagyfelbontású grafika forgatását a SYS(W) utasítással indíthatjuk el. A grafikát a SYS(W+124)-gyel lehet be-, illetve a SYS(W+141) utasítással pedig kikapcsolni. A W-nek más kezdőértéket adva lehet a forgató programot más tárterületre betölteni a BASIC betöltővel. Ha a kazettás magnót nem használjuk, akkor a gépi kódú program számára jó elhelyezést nyújt a kazettapuffer is: W=828.

Ha a Simon's BASIC által létrehozott ábrákat kívánjuk forgatni, akkor az 1. lista 1090-es sorában lévő $20-at kell módositanunk $E0-ra, mivel a Simon's BASIC $E000-tól kezdődően helyezi el a grafikus ábrákat. A 2. listán látható BASIC betöltő programnak az 1020-as sorában található adatát, a 32-t kell módositanunk 224-re.

Ha nem kívánjuk a teljes képernyőt forgatni, hanem annak csak egy részletét, akkor az 1. lista 1090-es, 1100-as és 1110-es soraiban kell végrehajtani módosításokat (illetve a 2. lista 1020-as sorában kell a megfelelő adatokat megváltoztatnunk).

Például az 1. listában módosítva az alábbi sorokat:

```
1090 KEP     .BYTE $20, $28
1100 OSZLOP.BYTE 18
1110 SOR     .BYTE 10
```

már csak a képernyő egy kisebb részlete forog majd.

A 2. listában hasonlóan módosítható az 1020-as sor:

1020 DATA 32,40,18,10,24,7,7,0,0,0

ilyenkor természetesen az 1030-as sorban található ellenőrző összeget is módositanunk kell 133-ról 138-ra. Közvetlen üzemmódból is végrehajthatjuk ezeket a módosításokat. Így például a

POKE 679,0:POKE 680,224

utasításokkal a forgatandó képernyő-területet $E000-tól kezdődően jelölhetjük ki; a

POKE 681, OSZLOP

a forgatásban résztvevő karakteroszlopok számát határozza meg: ilyenkor (OSZLOP+1) számú karakteroszlopnak megfelelő memóriaterület vesz részt a forgatásban. A

POKE 682, KARSOR

a forgatásban résztvevő KARSOR+1 számú karaktersor számát határozza meg. Az OSZLOP, illetve a KARSOR változókra a következő egyenlőtlenségeknek kell teljesülniük:

0 <= OSZLOP <= 39
0 <= KARSOR <= 24

Felhivom a figyelmet arra, hogy ezekkel a módosításokkal óvatosan kell bánnunk, ugyanis tekintettel kell lennünk arra is,

```
                        ; EZ A PROGRAM A NAGYFELBONTASU
                        ; KEPERNYOT FORGATJA BAJTONKENT
                        ; GRAFIKUS KEPERNYO HELYE - $2000
                        ; GRAFIKA BE - SYS(52604)
                        ; GRAFIKA KI - SYS(52621)
                        ; FORGATAS   - SYS(52480)
                        ;
150:    CD00                    *=  $CD00
170:    CD00            AKT     =   $FB
180:    CD00            KOV     =   $FD
                        ;
200:    CD00 A5 01      KEZD    LDA  #1
210:    CD02 8D B0 02           STA  TEMP+2
220:    CD05 A9 34              LDA  ##34
230:    CD07 78                 SEI             ; IRQ TILOS
240:    CD08 85 01              STA  #1         ; RAM
260:    CD0A AD A7 02           LDA  KEP
270:    CD0D 8D AE 02           STA  TEMP
280:    CD10 AD A8 02           LDA  KEP+1
290:    CD13 8D AF 02           STA  TEMP+1
310:    CD16 AD AA 02           LDA  S0R
320:    CD19 8D AB 02           STA  S0RSZM
350:    CD1C AD AC 02 KEPF0RG   LDA  KARAK
360:    CD1F 8D AD 02           STA  KARBJT
390:    CD22 AE A9 02 KARS0R    LDX  OSZLOP
410:    CD25 18                 CLC
420:    CD26 AD AE 02           LDA  TEMP
430:    CD29 85 FB              STA  AKT
440:    CD2B 69 08              ADC  #8
450:    CD2D 85 FD              STA  KOV
460:    CD2F AD AF 02           LDA  TEMP+1
470:    CD32 85 FC              STA  AKT+1
480:    CD34 69 00              ADC  #0
490:    CD36 85 FE              STA  KOV+1
510:    CD38 A0 00              LDY  #0
520:    CD3A B1 FB              LDA  (AKT),Y
530:    CD3C 48                 PHA
550:    CD3D B1 FD      GRAFS0R LDA  (KOV),Y
560:    CD3F 91 FB              STA  (AKT),Y
570:    CD41 18                 CLC
580:    CD42 98                 TYA
590:    CD43 69 08              ADC  #8
600:    CD45 AB                 TAY
610:    CD46 90 04              BCC  TOVABB
620:    CD48 E6 FC              INC  AKT+1
630:    CD4A E6 FE              INC  KOV+1
640:    CD4C CA         TOVABB  DEX
650:    CD4D D0 EE              BNE  GRAFS0R
670:    CD4F 68                 PLA
680:    CD50 91 FB              STA  (AKT),Y
710:    CD52 EE AE 02           INC  TEMP
720:    CD55 D0 03              BNE  TOV2
730:    CD57 EE AF 02           INC  TEMP+1
740:    CD5A CE AD 02 TOV2      DEC  KARBJT
750:    CD5D 10 C3              BPL  KARS0R
780:    CD5F 18                 CLC
790:    CD60 AD AE 02           LDA  TEMP
800:    CD63 69 38              ADC  #56
810:    CD65 8D AE 02           STA  TEMP
830:    CD68 AD AF 02           LDA  TEMP+1
835:    CD6B 69 01              ADC  #1
837:    CD6D 8D AF 02           STA  TEMP+1
850:    CD70 CE AB 02           DEC  S0RSZM
860:    CD73 10 A7              BPL  KEPF0RG
880:    CD75 AD B0 02           LDA  TEMP+2
890:    CD78 85 01              STA  #1
900:    CD7A 58                 CLI
910:    CD7B 60                 RTS
                        ;
930:    CD7C AD 11 D0 GRAFBE    LDA  $D011
940:    CD7F 09 20              ORA  ##20
950:    CD81 8D 11 D0           STA  $D011
960:    CD84 AD 18 D0           LDA  $D018
970:    CD87 09 08              ORA  ##08
980:    CD89 8D 18 D0           STA  $D018
990:    CD8C 60                 RTS
                        ;
1010:   CD8D AD 11 D0 GRAFKI    LDA  $D011
1020:   CD90 29 DF              AND  ##DF
1030:   CD92 8D 11 D0           STA  $D011
1040:   CD95 AD 18 D0           LDA  $D018
1050:   CD98 29 F7              AND  ##F7
1060:   CD9A 8D 18 D0           STA  $D018
1070:   CD9D 60                 RTS
                        ;
1085:   02A7                    *=   $02A7
1090:   02A7 00 20      KEP     .BYTE$00,$20
1100:   02A9 27         OSZLOP  .BYTE39
1110:   02AA 18         S0R     .BYTE24
1130:   02AB 18         S0RSZM  .BYTE24
1140:   02AC 07         KARAK   .BYTE7
1145:   02AD 07         KARBJT  .BYTE7
1150:   02AE 00 00 00   TEMP    .BYTE0,0,0
```

1. lista

```
1000 FOR I=679 TO 688
1010 READ X : POKEI,X : S=S+X: NEXT
1020 DATA 0,32,39,24,24,7,7,0,0,0
1030 IFS<>133THENPRINT"HIBAS ADAT !":END
1040 S=0
1050 W=52480: FOR I=W TO W+157
1060 READ X : POKEI,X : S=S+X: NEXT
1070 DATA 165,1,141,176,2,169,52,120
1080 DATA 133,1,173,167,2,141,174,2
1090 DATA 173,168,2,141,175,2,173,170
1100 DATA 2,141,171,2,173,172,2,141
1110 DATA 173,2,174,169,2,24,173,174
1120 DATA 2,133,251,105,8,133,253,173
1130 DATA 175,2,133,252,105,0,133,254
1140 DATA 160,0,177,251,72,177,253,145
1150 DATA 251,24,152,105,8,168,144,4
1160 DATA 230,252,230,254,202,208,238,104
1170 DATA 145,251,238,174,2,208,3,238
1180 DATA 175,2,206,173,2,16,195,24
1190 DATA 173,174,2,105,56,141,174,2
1200 DATA 173,175,2,105,1,141,175,2
1210 DATA 206,171,2,16,167,173,176,2
1220 DATA 133,1,88,96,173,17,208,9
1230 DATA 32,141,17,208,173,24,208,9
1240 DATA 8,141,24,208,96,173,17,208
1250 DATA 41,223,141,17,208,173,24,208
1260 DATA 41,247,141,24,208,96
1270 IFS<>19009THENPRINT"ADAT HIBA!":END
1280 PRINT"RENDBEN!":END
```

2. lista

```
 1 REM *** SPRITE ADATOK BEOLVASASA.
 3 FORI=0TO62:READA:POKE704+I,A:NEXT
 5 V=53248:POKE2040,11:POKEV+21,1
 7 POKEV,255:POKEV+1,80
 9 POKEV+23,1:POKEV+29,1
11 :
13 SYS52604 :REM  GRAFIKA BE
15 REM A PONT FEHER, A HATTER FEKETE.
17 FORI=1024TO2023:POKEI,16:NEXT
19 FORI=8192TO16191:POKEI,.:NEXT
21 BASE=8192:GOSUB45
23 BASE=8392:GOSUB45
25 BASE=11392:GOSUB45
27 BASE=11592:GOSUB45
29 BASE=14592:GOSUB45
31 BASE=14792:GOSUB45
33 REM *** GRAFIKA FORGATASA ***
35 FORI=0TO79:SYS(52480):NEXT
37 GOSUB63:REM GRAFIKA KI
39 END
41 :
43 REM SPRITE ATMASOLASA A GRAFIKABA
45 RESTORE
47 FORY=0TO20
49 FORI=0TO2
51 READ AX:X=I*8
55 BYT=BASE+320*INT(Y/8)+(YAND7)+X
57 POKE BYT,AX
59 NEXT:NEXT:RETURN
61 :
63 GETA$:IFA$=""THEN63
65 POKEV+21,0:PRINT"":REMSPRITE KI
67 SYS52621:REM GRAFIKA KI
69 RETURN
71 :
73 REM * A SZELLEMGRAFIKA ADATAI *
75 DATA 0,0,0,0,15,128,0,48,112,0,64,8
77 DATA0,128,4,0,128,4,1,4,18,127,36,18
79 DATA192,4,17,128,4,17,128,4,16,128
81 DATA4,16,128,4,16,128,196,16,71,130
83 DATA32,63,193,193,24,96,6,0,48,24,0
85 DATA16,96,0,25,128,0,14,0
```

3. lista

hogy a forgatás ne terjedjen ki olyan memóriaterületekre is, melyek már nem tartoznak a nagyfelbontású képhez. Ez az eset áll elő akkor, ha csak a KEP-hez tartozó bájtok értékét növeljük meg $00, $20-ról például $00, $30-ra. Ebben az esetben hiába toltuk fel korábban a BASIC munkaterület kezdetét 16 kbájtra; mert a forgatás hatása ki fog terjedni a $4000 feletti memóriaterületre is (!?), tönkretéve ily módon az ott elhelyezett BASIC programunkat is!

Lévén, hogy a felforgatott memóriaterület ilyenkor $3000 — $4F3F-ig, decimálisan: 12288-tól 20287-ig (12 kbájttól majdnem 20 kbájtig) terjed. Ha a SYS(W+124) utasítással kapcsoltuk be a grafikus képernyőt, akkor ilyenkor mi nem is látjuk a teljes memóriaterület forgását, mert a SYS(W+124) úgy kapcsolja be a nagyfelbontású képernyőt, hogy az $2000-től $3F3F-ig (8192-től 16191-ig) terjed. Emiatt a nagyfelbontású képernyő felső fele: $2000 — $2FFF-ig változatlanul marad; $3000-től a képernyő végéig: $3F3F-ig látjuk a forgást, de az e terület fölötti $3F40-től $4F3F-ig terjedő memóriaterület forgása előttünk is rejtve marad. Illetve csak közvetve tapasztalhatjuk a forgatás eredményét úgy, hogy a 16 kbájt fölött ($4000-től) elhelyezkedő BASIC programunkat a forgató rutin jól kiforgatja eredeti formájából!

A grafikus kép a képernyőről „kiúsztatható", ha az 1. lista 520-as sorában található LDA (KOV), Y-t megváltoztatjuk LDA #$00-ra. Ez a

POKE W+58,169 : POKE W+59,0

utasítások segítségével közvetlen üzemmódból is megvalósítható. Ezután meghíva a forgató rutint, az törli a grafikus kép utolsó karakteroszlopának memóriatartalmát. (Próbáljuk ki bekapcsolt grafikai képernyővel a

FOR I=0 TO 39 : SYS(W) : NEXT

utasítást!)

Ha elhagyjuk az 1. lista 510—530-as, valamint a 670—680-as sorait, vagy az itteni utasításokat NOP-pal (kódja: $EA=234) helyettesítjük, akkor akár egy grafikai képújságot is megvalósíthatunk. Ilyenkor ugyanis a grafikus képernyő legutolsó karakteroszlopának memóriatartalmát nem írja felül a képernyő legelső karakteroszlopának memóriatartalma, azaz ilyenkor nem jön létre a grafikus kép körbeforgatása, hanem csak a karakteroszlopok memóriatartalma egy oszloppal előrébb másolódik át. Ilyen módon tehát, ha a legutolsó karakteroszlopba, minden forgatás előtt újabb képi információt írunk be, akkor egy grafikus képújságot valósíthatunk meg.

Végezetül pedig a *3. listán* egy BASIC példaprogram található, illusztrálandó a nagyfelbontású képernyő forgatását. (Figyelem! A gépi kódú programot ez előtt a program előtt a C64 memóriájába kell tölteni!) Ez a BASIC program először egy sprite-ot jelenít meg kétszeres nagyításban (1-9-es sor), majd ezt a szellemgrafikát bemásolja a nagy felbontású képernyő különböző helyeire (21-31-es sor). Az így keletkezett grafikus képet 80 alkalommal balfelé lépteti egy karakternyivel, azaz a képet kétszer teljesen körbeforgatja! Figyeljük meg, hogy a kinagyított sprite eközben a helyén marad. Ez a kinagyított sprite azért nem vesz részt a nagyfelbontású képernyő forgatásában, mert nem tartozik hozzá a nagyfelbontású képernyő memóriatartalmához.

A sprite-adatokat természetesen további sprite-adatokkal bővíthetjük ki, és így azok is a nagyfelbontású képbe másolhatók át, például úgy, hogy az átmásoló alprogramot nem GOSUB45-tel, hanem GOSUB47-tel hívjuk meg.

A nagyfelbontású képernyő bájtonkénti balra forgatásának ismertetésével be is fejeztük a forgatásokról szóló sorozatunkat.

SZABÓ PÉTER PÁL

Környezetvédő „ZÖLD" rutinok, avagy a célzott szemétgyűjtés

A Mikroszámítógép Magazin 1988/1. számában megjelent a CHIP újságnak Tóth Ferenc fordításában és kiegészítésében a Szemétgyűjtés című cikke. Az abban ismertetett eljárás helyett egy egyszerűbb, mégis hatékonyabb módszert szeretnék ajánlani a programozóknak.

Az említett cikk a sztringmemória-terület gyors nagytakarításáról egy gépi kódú szubrutinnal gondoskodik, amely „sokkal gyorsabb, mint a C64 rendszerrutinja".

A szubrutin természetesen működik, és ellátja feladatát. Van viszont néhány kellemetlen vonzata:

— a sztringértékadásokat tartalmazó programokban *minden* füzérértékadás előtt meg kell hívni a rutint,

— maximum 2048 sztringet tud kezelni,

— ha a rutin szemetet talál a sztringterületen, akkor az összes ide mutató sztringtömb és sztringváltozó címét áthelyezi az operatív memória $F000—$FFFF területére, amit mi esetleg más célra szeretnénk használni,

— adott esetben a szemét nagytakarításának néhány másodperces ideje is tűrhetetlenül hosszú,

— maga a rutin is helyet foglal a memóriában, ami BASIC-bővítő programok alkalmazásánál szintén problémákat okoz.

E hátrányokat kiküszöböli a következő, lényegesen egyszerűbb és csak néhány BASIC-utasításból álló szemétgyűjtő módszer, amelynek lényege, hogy *csak ott gyűjt szemetet, ahol van!*

Igen ám, de ehhez tudnunk kell a szemét keletkezéséről. Márpedig szemét akkor keletkezik, ha egy sztring új értéket kap, mert a régi feleslegessé válik. A C64 Garbage Collection rendszerrutinja aktiválódásakor a teljes sztringterületet (a BASIC-terület végétől — S37/S38 — lefelé, a S33/S34 által mutatott értékig) átfésüli szemét után kutatva, akkor is, ha például a sztringterület 90 százalékát egy vagy több változatlan sztringtömb tölti ki a sztringterület elején, amely természetesen szemétmentes.

E felesleges kutatás a legnagyobb időrabló. Mi hát a megoldás?

— A szemét keletkezése előtt jegyezzük fel a sztringterület végét jelző mutató értékét:

SA = PEEK(51) : SF = PEEK(52)

— A szemétgyártás végén, például formázott kiíratást — PRINTUSING-ot — megvalósító alprogram után állítsuk be a szemétmentes terület végére a BASIC-terület végét:

POKE 55,SA : POKE 56,SF és hívjuk meg a Garbage Collection rendszerrutint: SYS 46374. A rendszerrutin ekként csak azon a viszonylag szűk területen dolgozik, ahol a szemét koncentrálódott.

— Állítsuk vissza a BASIC-terület végét jelző mutatót az eredeti értékre: POKE 55,BA : POKE 56,BF. A BA és BF értéket a program elején jegyeztük fel:

BA = PEEK(55) : BF = PEEK(56).

A hatást mutatja be az *1.* és *2. listán* látható két demoprogram. A 2. lista tartalmazza a ZÖLD rutinokat, az 1. nem. Mindkét programban létrehozunk egy-egy nagy sztringtömböt (10-es sor) és 1 darab „szemét" sztringet. A 20-as sorban hívjuk meg a Garbage Collection rendszerrutint. A futási idők jól demonstrálják a módszer előnyét. A 2. ZÖLD rutint természetesen nem fontos közvetlenül a szemét keletkezése után mindig meghívni, megvárhatjuk, míg összegyűlik annyi, amennyinek gyűjtési ideje még elviselhető számunkra.

Érdemes figyelembe venni azt a gondolatot is, hogy *ha nincs szemét, nem kell gyűjteni!* Szemét csak sztringből lesz, célszerű ezért a programokban, ahol csak lehet, sztring helyett numerikus adatokkal dolgozni.

Ha ezt tesszük, és a ZÖLD rutinokkal célszerűen védjük programunk környezetét, az sohasem merevedik le, mert nem kell „nagytakarítást" csinálni, azaz elérjük célunkat: programunkat még másodpercekig tartó szünetek sem zavarják, legfeljebb tizedmásodpercesek, amelyeket szinte észre sem lehet venni.

KÁNTOR JÓZSEF

1. lista

```
T=345.43333s


0 REM------ DEMO 1 ------------------
5 DIMA$(2050)
10 FOR I=1TO 2048:A$(I)=CHR$(65):NEXT
15 B$=STR$(1):B$=STR$(2)
20 TT=TI:GOSUB50:PRINT"T=";(TI-TT)/60;"SEC"
30 REM
40 REM
50 SYS46374:RETURN:REM---GARBAGE COLLECTION--
```

2. lista

```
T=0.35s

0 REM------ DEMO 2 ------------------
2 BA=PEEK(55):BF=PEEK(56):REM-BASICHATAR
5 DIMA$(2050)
10 FOR I=1TO 2048:A$(I)=CHR$(65):NEXT
11 GOSUB40:REM--ELSO SZEMETMENTES--
15 B$=STR$(1):B$=STR$(2)
20 TT=TI:GOSUB50:PRINT"T=";(TI-TT)/60;"SEC"
30 REM
32 REM
40 SA=PEEK(51):SF=PEEK(52):RETURN:REM--ZOLD1-
42 REM
44 REM
50 POKE55,SA:POKE56,SF:SYS46374
55 POKE55,BA:POKE56,BF:RETURN:REM------ZOLD2-
```

MONITOROLÓGIA

Mit kell tudnunk a monitorokról?

Akik gyakran ülnek számítógépük előtt programozás vagy játék céljából, jól tudják, hogy a megjelenítésre használt fekete-fehér vagy színes televízió mennyire fárasztja a szemet. Az igényesebb felhasználók ezért jobb felbontású és élesebb képet szeretnének. A problémát monochrom vagy színes monitor vásárlásával igyekeznek megoldani. Ha azonban nem körültekintően vásárolnak, még több problémával találkoznak. Az alábbiakban a monitor kiválasztásában, a különböző paraméterek és jelek, valamint a csatlakozók sokaságában kívánunk eligazítást nyújtani.

A különböző fogalmak jobb megértéséért először röviden a fekete-fehér tévékép előállításának elvét tekintsük át. A képet a fluoreszkáló képernyőn egy elektronsugár állítja elő. A kép pontokból áll. Az egyes pontok fényességét az elektronsugár erőssége szabja meg. Az elektronsugár eltérítését két egymásra merőleges, változó intenzitású mágneses tér végzi az *1. ábrán* látható módon. A színes kép előállítása ettől annyiban tér el, hogy a vörös, zöld, kék alapszíneknek megfelelő három elektronsugár intenzitásainak aránya határozza meg egy pont színét. Ennek megfelelően egyidejűleg három elektronsugár eltérítéséről kell gondoskodni . . .

. . . DE NEM TERRORRAL

A sugár a képernyő bal felső sarkából indul és jobbra mozog, miközben egy kicsit lefelé is elmozdul, és így leír egy sort. Ezután a sugár visszaugrik az előző kiindulási pont alá, és újabb sort ír le. Ez mindaddig ismétlődik, amíg a sugár a képernyő jobb alsó sarkát el nem éri. Ezután a sugár visszaugrik a bal felső sarokba, és a folyamat az előbb leírt módon ismétlődik. Az elektronsugár visszaugrása alatt az elektronsugarat kikapcsolják, hogy ilyenkor ne lehessen látni. Azt, hogy a képernyőn egyszerre a teljes képet látjuk, a képernyő utánvilágításának és a szem tehetetlenségének köszönhetjük.

Utánvilágításnak azt a jelenséget nevezzük, hogy az elektronsugár által gerjesztett pont rövid ideig azután is világít, miután az elektronsugár azt elhagyta. A nálunk érvényes televíziószabványban a sorok száma 625, a sorfrekvencia (a másodpercenként lerajzolt sorok száma) 15 625 Hz. Ezekből 25 Hz-es képfrekvencia (a másodpercenként előállított képek száma) adódik, amely a szem számára a folyamatos változás benyomásának keltésére már elegendő (lásd például a mozgófilmet). Ezzel szemben a televízió technikájában a villódzás elkerülésére trükköt alkalmaznak. A teljes képet két fél képre osztják, és az egyes fél képeket kétszeres sortávolságban egymás után rajzolják fel úgy, hogy a két fél kép egymásba ágyazódik. Ezt nevezik váltott soros letapogatásnak. Mindegyik fél kép 312,5 sorból áll, és a félkép-váltási frekvencia 50 Hz.

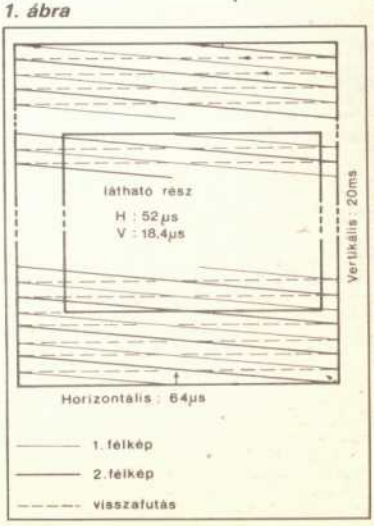
Az elektronsugár mozgatását végző mágneses teret a vízszintes és függőleges eltérítőtekercsek állítják elő, amelyeket egy 15 625 Hz-es és egy 50 Hz-es szabadon futó oszcillátor vezérel. E két oszcillátor frekvenciája együttesen határozza meg az elektronsugár helyzetét. Ahhoz, hogy egy bizonyos képinformációt egyértelműen a képernyő egy megadott pontjához lehessen rendelni, a két oszcillátort szinkronizálni kell. Ezt akkor hozzuk létre, amikor az elektronsugár a sor végéről a következő sor elejére, vagy a kép jobb alsó sarkából a bal felső sarkába ugrik visszsa, mivel ezalatt az elektronsugarat „feketére kapcsoljuk", vagyis kioltjuk, ezáltal a szinkronizáló jelek nem zavarják a képet.

A képinformáció, a kioltó és a szinkronizáló jelek együttesen alkotják az összetett videojelet, idegen röviditéssel BAS-jelet, amely a kép egyértelmű előállításához szükséges. Az összetett videojel felépítését és a jelszinteket a *2. ábra* mutatja.

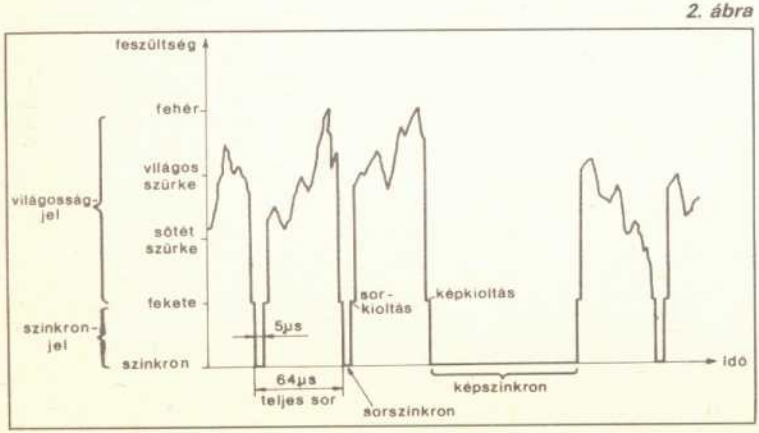
TEMESSÜK A VILLÓDZÁST!

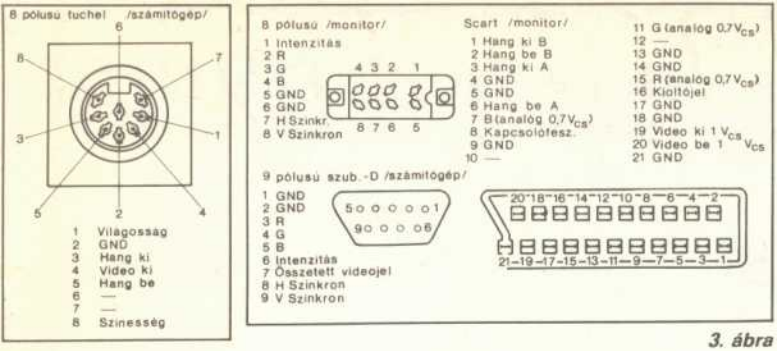
Egy sor lerajzolásának ideje H = 64 µs, egy fél kép előállításáé pedig V = 20 ms. Monitoroknál nem törekednek arra, hogy a teljes képet lássuk, ezért csak az 1. ábrán bekeretezett kisebb részük látható. Alapvető követelmény viszont, hogy a kontraszt jó legyen, a grafikus megjelenítéskor a legfi-

1. ábra



2. ábra





3. ábra

nomabb vonal se veszítsen a világosságából. Tételezzük fel, hogy vizszintesen 640 pontot jelenítünk meg (80 karakter × 8 pont), akkor az időbeli különbség két pont között 78 ns-ra adódik, ha a látható ablak szélességét az 1. ábrán megadott 52 µs-ra vesszük fel. Ez az érték pedig éppen 13 MHz sávszélességet határoz meg, amit már a jobb televíziók videobemenetei sem tudnak teljesíteni. Ha figyelembe vesszük, hogy ez a sávszélesség a nagy felbontású grafikához igen kevés, továbbá némi tartalékra is szükség van, valamint a látható ablak gyakran kisebb is, akkor a 18 MHz-es sávszélesség is valójában csak rossz kényszermegoldás. Mivel kizárólag statikus képet ábrázolunk, a váltott soros letapogatásnál a szokásos eltérítési frekvenciáknál zavaró villódzás keletkezik. Ez háromféle módon küszöbölhető ki:

— A képernyő utánvilágítási idejének növelésével. Az utánvilágítási idő szokásos értéke: 12 . . . 22 ms és tetszőlegesen nem növelhető, mert scrollnál zavaró lehet. Bizonyos esetekben 54 ms-ot is alkalmaznak. Ez az érték azonban már az elviselhetőség határán van.

— Egyszerűen elhagyják a váltott soros letapogatást, és mindig csak az egyik fél képet jelenítik meg. A sor- és képfrekvencia azonos marad, a kép nyugodtabb lesz, és kiegyenlítő impulzusokra nincs szükség. Némi manipulációval a 312 sorból 300 láthatóvá tehető. Ha több sorra van szükség, akkor megnövelik a sorfrekvenciát, aminek következtében a szükséges video-sávszélesség azonos módon megnövekszik. Néhány számítógép lehetővé teszi a váltott soros letapogatás be- és kikapcsolását (interlaced/non-interlaced mode).

— Megnövelik a képfrekvenciát. Ennek következtében a függőleges felbontás csökken, ami a sorfrekvencia növelésével egyensúlyozható, de ehhez az imént említett nagyobb video-sávszélesség kell.

Túlnyomórészt szöveges megjelenítésnél a második módszerben leírtakat célszerű alkalmazni, grafikus ábrázolásnál pedig a másik két módszert. Természetesen semmi akadálya sincs több módszer egyidejű alkalmazásának.

A fentiek alapján megadhatók a „szokásos" BAS-kimenetű számítógépekhez csatlakoztatható monitorok főbb paraméterei:

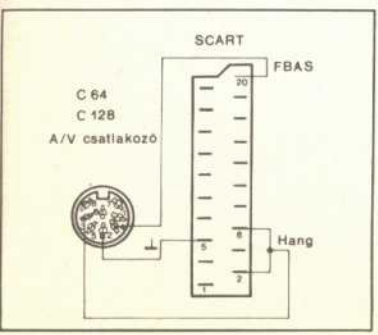
— sávszélesség: nagyobb, mint 18 MHz,
— sorfrekvencia: 15,5 . . . 18,5 kHz,
— képfrekvencia: 50 . . . 60 Hz,
— utánvilágítási idő: 12 . . . 22 ms.

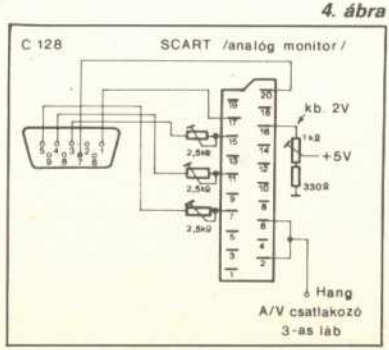
Célszerű olyan monitort választani, amelynél a kép- és sorfrekvencia kivülről forgatógombbal állítható (a monitorok többsége ilyen). Vásárlás előtt mindenképpen ajánlatos a számítógép kimenetét alaposan tanulmányozni, mert az egyes gépek kimenetei a fenti értékektől jelentősen eltérhetnek. Jó példa erre az Atari ST; ennek a sorfrekvenciája 36 kHz és a képfrekvenciája 71 Hz.

SZIGORÚ JELLELTÁR

Eddig minden világos lenne, ha a BAS-csatlakozáson kívül nem létezne a TTL-csatlakozások széles skálája. Utóbbiak-

4. ábra





nak közös tulajdonsága, hogy a video- és szinkronjeleket külön vezetékeken továbbítják, és legalább ez utóbbiak TTL-szintűek. Ezért meg kell állapítani, hogy a számítógéptől jövő szinkronjelek külön-külön vagy keverve jönnek-e, illetve melyik az aktív szintjük. Gyakran ugyanannak a számítógépnek a különböző videokártyái is más-más jeleket szolgáltatnak.

A szinkronjelekkel kapcsolatban itt két „apróságra" érdemes ügyelni. Van, amikor a horizontális szinkronjelet a vertikális szinkronjellel egyszerűen VAGY kapcsolatba hozzák. Ezt a monitor még „megérti", de a vertikális szinkron alatt a horizontális szinkron kiesik, és a kép kezdetén csak lassan áll vissza. Ebben az esetben az első szövegsorban vagy az első grafikus sorokban „elkenődéseket" láthatunk. A másik változatban a horizontális és vertikális szinkronjeleket KIZÁRÓ VAGY kapcsolatba hozzák, ami a szabványjelhez közelebb álló. Ilyenkor a felső képszél villódzhat. Ha a két kisebb galiba valamelyike előfordul, ez nem a mi készülékünk hibája, ezért csupán a kényelmes fejlesztőket kell szidni.

Akinek még ez sem elég, az vásároljon színes monitort, amelynél az egyszínű monitorok „szabvány összevisszaságaihoz" még újabbak társulnak. Ezek közül csak a két legfontosabbat említjük meg: a TTL és az analóg RGB rendszer által okozott problémákat. A különbség első ránézésre a három videojel (RGB) szintjeiben van:

analóg RGB: 0,7 . . . 1 V
TTL RGB: 3 . . . 3,5 V

Lényegesebb különbséget jelent, hogy a TTL RGB jel bináris, tehát a fehéret és feketét is beleértve összesen nyolc szín ábrázolható azonos intenzitással. Ezen úgy segítenek, hogy még egy intenzitásbitet fűznek hozzá (16 szín). Ezzel szemben az analóg RGB a vörösből, a zöldből és a kékből keverhető összes színt tetszőleges intenzitással elő tudja állítani.

A számítógép és a monitor közötti szintproblémák előtét-ellenállásokkal könnyűszerrel áthidalhatók.

A fentieknél is változatosabb képet nyújtanak azonban a számítógépek és a monitor csatlakozói. Ha netán a vásárlásnál az előbbi dolgok alapos tanulmányozása miatt a megfelelő csatlakozókábel vásárlásáról megfeledkeznének, a *3. ábrán* megadom néhány gyakori csatlakozó bekötési rajzát. A *4. ábrán* két összekapcsolási példát is láthatunk Commodore 64 és 128 típusú számítógépekre. Természetesen ezen a téren teljességre nem törekedhettem a változatok nagy száma miatt, de úgy érzem, hogy a bemutatott példák támpontul szolgálhatnak az egyes egyedi esetek megoldásához. A szakirodalomban egyébként nagyon sok kézenfekvő megoldás található. Az összekapcsoláshoz azonban mindig kérjük ki szakember tanácsát, ha nem akarunk kép helyett „salátát" látni.

DR. MENTES GYULA

Boldogok, akik futják a köröket, mert fordulatos az életük

MÓKUSKERÉK

Azt a jelenséget, amikor egy új szubrutint ír valaki, pedig egy azonos célú rutin már ott lapul a programkönyvtárban, a „kerék feltalálásának" nevezik. Az ilyenkor alkalmazott megoldások erős nyelvfüggőséget mutatnak, amit néhány tipikus példával illusztrálunk:

— Az assembly programozók milliói számára készítenek kereket, de nincs közöttük kettő, amelyik közös tengelyre illene.

— Az Ada-programozók szögletes kereket terveznek, majd pedig olyan utakat specifikálnak, melyeknek gödreibe azok beleillenek.

— Az ALGOL-programozók tartózkodnak a kerék készítésétől, mert azok esetleg go to valahova.

— Az APL-programozók a kereket a levegőben, egy zuhanó repülőgép futóművén lógva építik, és földet érés előtt általában már majdnem készen is vannak.

— A BASIC-programozók két kézzel sem találják a tengelyüket.

— A C-programozók kerék helyett propellert készítenek. Ha kereket is tudnának, legalább „B minusz" osztályzatuk lenne.

— A COBOL-programozók EZER-KEREKŰ-SZÁLLÍTÓ-MODUL-t dolgoznak ki, és megtiltják a gyaloglást.

— A FORTRAN-programozók az őrület határára jutnak abbeli igyekezetükben, hogy I-vel kezdődő kerekeket állítsanak elő.

— A FORTH-programozók számlálatlanul gyártják a kerekeket, majd elfelejtik, melyik veremben tárolták őket.

— A LISP-programozók kidolgozzák a kerekeket (a kerékben (a kerékben (a programozó kerekében)))(újabban már autójuk is van).

— A LOGO-programozóknak nem kell kerék, nekik teknőcük van.

— A Pascal-programozók az erény szintjére emelik a gyaloglást.

— A PL/I-programozók kijelölnek egy teamet, amelyik megépíti azt a gyárat, ahol majd a mérethibás kerekeket előállítják.

— A PROLOG-programozók által készített dolgok körbe-körbe járnak ugyan, de nem jutnak sehová.

— Az RPG-programozók soha nem készítenek kereket. Egy vezérlésre alkalmas kormánykeréktől eltekintve már minden kerekük megvan.

— A Smalltalk-programozók a kerekeket beszélik rá önmaguk felépítésére.

— A SNOBOL-programozók sztringeken működő kerekeket hoznak létre. Ezeket köznyelven jojonak hívják.

(Zeke Hoskin)

(Megjelent az APL Quote–Quad 1988. júniusi számában)

C & ZX

SOFTWARE BÖRZE

folyamatosan bővülő programválasztékkal,
kazettán és lemezen egyaránt.

***

COMMODORE 64, SINCLAIR ZX SPECTRUM
játékleírások, térképek, ZX POKE-ok,
cheat-ek, keresztrejtvény nyeremények,
+4 oldal ENTERPRISE melléklet.

SPECTRUM VILÁG

megjelenik havonta 11000 példányban,
terjeszti a Magyar Posta,
kapható a hírlapárusoknál!

***

SEIKOSHA SP–1200 VC

Ma még bizonyára kevesen ismerik Magyarországon ezt a Nyugaton már igen elterjedt nyomtatótípust. Azért ajánlom az olvasók figyelmébe, mert előnyös tulajdonságaival messze felülmúlja az MPS 803-ast, amellyel felülről kompatibilis. Leginkább a CITIZEN 120D típusú nyomtatóval lehetne összehasonlítani, hiszen kevés eltéréssel ugyanolyan lehetőségeket kínál.

A gépbe éppúgy befűzhetünk perforált papírt, mint bármilyen más minőségű, A/4-esnél nem nagyobb lapot. Sőt be sem kell fűznünk, csak a vezetőtámlára helyezve fixáljuk. A befűzés és beállítás már automatikus.

A nyomtatást kilenc tű, 120 karakter/másodperc sebességgel végzi. Mivel a fejet függőlegesen elmozdíthatjuk, lehetőség nyílik közel levél minőségű betűk nyomtatására is (NLQ). Azt is kiválaszthatjuk, hogy vízszintesen hány karaktert írjon egy oldalra (pl. 40,

2. ábra

Ø123456789abcdefghijklmnopqrstuvwxyz

Ø123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ø123456789ABCDEFG

Ø123456789abcdefg

Ø123456789ABCDEFG

Ø123456789abcdefg

3. ábra

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFG

Ø123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ø123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ø123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

Ø123456789ABCDEFGHIJKLMNOPQRSTUVWXYZ

1. ábra

```
CHR$(14).....DUPLASZELES NYOMTATAS BE
CHR$(15).....DUPLASZELES NYOMTATAS KI
CHR$(27);"-1".....ALAHUZAS BE
CHR$(27);"-Ø".....ALAHUZAS KI
CHR$(18).....INVERZ NYOMTATAS BE
CHR$(146).....INVERZ NYOMTATAS KI
CHR$(27);CHR$(1Ø8);CHR$(n).....BAL MARGO BETOLASA
CHR$(27);"Q";CHR$(n).....JOBB MARGO BETOLASA
CHR$(27);"E".....VIZSZINTES DUPLAZAS BE
CHR$(27);"F".....VIZSZINTES DUPLAZAS KI
CHR$(27);"G".....FUGGOLEGES DUPLAZAS BE
CHR$(27);"H".....FUGGOLEGES DUPLAZAS KI
CHR$(16);"n".....KURZOR POZICIO
CHR$(27);"4".....DOLT BETUS NYOMTATAS BE
CHR$(27);"5".....DOLT BETUS NYOMTATAS KI
CHR$(8).....GRAFIKUS UZEMMOD BE
CHR$(15).....GRAFIKUS UZEMMOD KI
CHR$(27);"3".....SORSURUSEG
CHR$(27);"SØ".....FELSO NYOMTATAS
CHR$(27);"S1".....ALSO NYOMTATAS
CHR$(27);"T".....NORMAL NYOMTATAS
CHR$(27);"X1".....IROGEP BETUMINOSEG BE
CHR$(27);"XØ".....IROGEP BETUMINOSEG KI
CHR$(27);"M".....SURTETT BETU BE
CHR$(27);"@".....SURITETT BETU KI
OPEN 13,4,13:PRINT#13.....BETUSURUSEG DUPLAZAS
CLOSE 13.....NORMAL BETUSURUSEG
CHR$(12).....LAPDOBAS
```

4. ábra

48, 80, 96, 160, 192) *(1. ábra)*. A függőleges sűrűség megválasztásakor a sorok távolságát kell megadni N/216 collban (0–255). Így természetesen a sorok fedhetik is egymást. Mindemellett többféle betűtípust állíthatunk be *(2. ábra)*.

További lehetőségek

A karaktereket megdönthetjük, aláhúzhatjuk, pontduplázást kérhetünk vízszintesen és függőlegesen is *(3. ábra)*. E műveletek a CHR$ kódokkal programozhatók *(4. ábra)*. Néhány funkciót a nyomtató kezelőszervéről is kiválaszthatunk: az írássűrűséget, az írógép-betűtípust stb. Ugyanitt kell kérni a lapdobást, a soremelést és az írásengedélyezését is. Mindezek állapotáról négy LED ad felvilágosítást: feszültség = zöld, betűtípus = zöld, papírállapot = piros, írásengedélyezés = zöld.

Szükség esetén nemcsak a CBM karakterkészlettel, hanem normál ASCII-ben is dolgozhatunk. Így grafikus üzemmódban akár 1920 pontos vízszintes felbontást is elérhetünk. CBM üzemmódban további választási lehetőség a standard és a din karakterkészlet.

A nyomtató hátlapján található kapcsolórendszer különböző, szoftverből nem módosítható funkciók beállítására szolgál *(5. ábra)* (például a nulla írását kérhetjük áthúzással vagy anélkül, megadhatjuk, hogy a lapvégérzékelő aktív vagy passzív állapotban legyen-e).

A nyomtató egyetlen hátránya, hogy a karakterek függőlegesen nem nagyíthatók, szemben a CITIZEN-nel, amelynél erre is van lehetőség.

BÁRTFAI BARNABÁS

5. ábra

N	FUNKCIO	BE	KI
1	Csatornaszam	5	4
2	Papirhossz	12"	11"
3	Karakterkeszlet 1.	ASCII	CBM
4	Karakterkeszlet 2.	DIN	STANDARD
5	Irasirany	CSAK ODA	ODA VISSZA
6	Papirveg erzekelo	NEM MUK.	MUKODIK
7	Nulla irasa	ATHUZVA	ATH. NEL
8	CSF	IGEN	NEM

Minden kedden 17-től 20 óráig
HCC ENTERPRISE klub
a VSZM
Közösségi Házban
(Bp. XI., Fehérvári út 120.)
Klubvezető: Romvári Gábor
Telefon: 810-950/473

Professional

Olvasóink tájékoztatására közöljük a Professional országos számítógépszerviz-hálózatának címeit.

1033 Budapest, Kaszásdülő u. 1—3.	Tel.: 805-587 805-565 805-278
3100 Salgótarján, Rákóczi út 252.	Tel.: 32-13-598
3526 Miskolc, Huba u. 23.	Tel.: 46-89-308
4400 Nyíregyháza, Mártírok tere 9.	Tel.: 42-14-032
4028 Debrecen, Besze J. u. 7.	Tel.: 52-25-687
5601 Békéscsaba, Tanácsköztársaság u. 75.	Tel.: 66-28-584
6701 Szeged, Retek u. 23.	Tel.: 62-25-448
7690 Pécs, Liceum u. 7.	Tel.: 72-33-955
7400 Kaposvár, Tóth L. u. 12.	Tel.: 82-12-104
8900 Zalaegerszeg, Bíró M. u. 14/A	Tel.: 92-13-789
9700 Szombathely, Rákóczi F. u. 50.	Tel.: 94-13-506
9023 Győr, Buda u. 34.	Tel.: 96-11-440
8000 Székesfehérvár, Móri u. 58.	Tel.: 22-16-814

ÁTALÁNYDÍJAS JAVÍTÁSI ÉS KARBANTARTÁSI SZERZŐDÉS = ÖRÖK GARANCIA

FOTOELEKTRONIK ALFA GT NOVOTRADE

COMMODORE, ATARI, TVC stb. személyi számítógépek valamint IBM PC/XT/AT professzionális PC számítógépek és perifériák (floppy, printer) garanciális és fizető JAVÍTÁSA, KARBANTARTÁSA.

SZERVIZEINK:

1053 Budapest V., Magyar u. 12–14.	T.: 173-551	Tx: 7621	4034 Debrecen, Holló László u. 14.		
1083 Budapest VIII., Szigony u. 9.	T.: 343-153		5600 Békéscsaba, Bartók Béla u. 37.	T.: 66-27-195	
1191 Budapest XIX., Gábor A. sétány 3.			6726 Szeged, Székelysor 13.	T.: 62-13-377	
3100 Salgótarján, Arany János u. 3.	T.: 32-14-007		7400 Kaposvár, Füredi u. 24.	T.: 82-16-307	
3525 Miskolc, Fazekas u. 1–3.	T.: 46-17-011		7624 Pécs, Jurisics M. u. 17.	T.: 72-11-812	
			9700 Szombathely, Szalonok u. 31.	T.: 94-14-519	

1053 Budapest, Henszlmann I. u. 9. Telefon: 174-144 Telex: 22-7621

VEVŐSZOLGÁLAT — FOTOELEKTRONIK — NOVOTRADE — ALFA G.T.

VÉTEL — ELADÁS 1077 Budapest, Dohány u. 16. Tel.: 428-936

POKE-kal vagy POKE nélkül?

```
DEFAULT     CHANNEL     TIMER
KEY         CLICK       RATE
DELAY       INTERRUPT   STOP
KEY         NET         CODE
TAPE        SOUND       PROTECT
LEVEL       FAST        SAVE
SOUND       STOP        BUFFER
SPEAKER     SERIAL      BAUD
FORMAT      VIDEO       MODE
COLOR       COLOUR      Y
X           BORDER      BIAS
STATUS      EDITOR      VIDEO
KEY         BUFFER      REM1
REM2        NET         CHANNEL
NUMBER      MACHINE     PRIORITY
ATTRIBUTES  PALETTE     COLOR
COLOUR      PAPER       INK
CHARACTER   CURSOR      CHARACTER
COLOR       COLOUR      SCROLL
ON          OFF         UP
DOWN        BEAM        ON
OFF         LINE        STYLE
MODE
```

1. lista

```
100 !SET ASK tovabbi vizsgalata
110 !
120 OUT 177,1:LET S=0
130 FOR I=21213 TO 21543 STEP 5
140    LET S=S+1
150    LET K=256*PEEK(I)+PEEK(I+1)-32768
160    LET H=PEEK(K):LET A$=""
170    FOR R=K+1 TO K+H
180       LET A$=A$&CHR$(PEEK(R))
190    NEXT
200    LET A$=A$&"                "
210    PRINT A$(0:11);
220    IF S=3 THEN
230       LET S=0:PRINT
240    END IF
250 NEXT
260 PRINT
```

2. lista

A C64-tulajdonosok a 2.0 verziójú BASIC szerény szolgáltatásain kívül a POKE utasítást igen gyakran kénytelenek használni. A számítógép-memória megfelelő helyére közvetlenül beírt értékekkel azután csodákat művelnek.

Az egyre magasabb verziószámú BASIC-ek egyre több utasítást tartalmaznak, és így a szín-, a hang- és a grafikus lehetőségek kihasználására új programozási módszerek kínálkoznak.

Az Enterprise BASIC-je hovatovább feleslegessé teszi a POKE utasítást — no azért el ne felejtsük! —, mert összetett kulcsszó-kombinációival lehetővé teszi szín-, hang-, rajz- és átdefiniálási feladatok megoldását.

```
C902  PUSH AF      ;allapot mentes
C903  LD HL,BFF2   ;most csak H kell
C906  LD A,C       ;A-ba EXOS valtozo szama
C907  CP 01        ;FLAG SOFT IRQ allitas?
C909  JR Z,C90E    ;akkor cimet nem bantunk
C90B  ADD A,C5     ;osszeallt a cim
C90D  LD L,A       ;most a cim BFC5+A
C90E  DEC B        ;mit csinaljak
C90F  JP M,C918    ; olvass
C912  JP Z,C917    ; irj
C914  LD A,(HL)    ; valts at
C915  CPL          ;komplemental
C916  LD D,A       ;beiras elokeszitese
C917  LD (HL),D    ;uj valtozo beiras
C918  LD D,(HL)    ;erteket visszaolvas
C919  POP AF       ;visszaallit
```

3. lista

```
100 !   karakter atalakito
110 ! SET CHARACTER n,BIN..... helyett
120 !
130 LET K=13440
140 PRINT "nyomd le azt a billentyut"
150 PRINT "amit at akarsz alakitani."
160 PRINT
170 LET A$=INKEY$
180 IF A$<" " THEN GOTO 170
190 PRINT A$;" ---> ";
200 LET KOD=ORD(A$):LET KEZD=K+KOD
210 FOR I=0 TO 8
220    READ MIT
230    SPOKE 255,KEZD+I*128,MIT
240 NEXT
250 PRINT CHR$(KOD)
260 DATA 62,65,85,65,73,65,93,65,62
```

4. lista

A SET, ASK kulcsszóval kezdődő utasítások természetesen nem csinálnak mást, mint az argumentumként megadott további információk alapján meghatározzák azt a — felülírható — memóriacímet, amelynek tartalmát meg kell változtatni a kívánt cél eléréséhez.

Az Enterprise tehát egy adott feladat elvégzésére általában több lehetőséget kínál. A programozó hangulatától, stílusától, korábbi beidegződéseitől függ, hogy melyik megoldást választja. Igaz, mindegyik programozási stílushoz meg kell tanulni a megfelelő kulcsszavakat, EXOS változó sorszámot és/vagy a memóriacímeket.

Természetesen nem kell ezeket az értékeket ész nélkül megjegyezni. Használjuk a gépkönyvet vagy a megfelelő segédletet, mert a bőség zavarában bizony nemegyszer nehéz eligazodni.

A SET, ASK, TOGGLE kulcsszavak után az interpreter például a következő kulcsszósorozatra, illetve azok kombinációjára végzi el az elemzést *(1. lista)*. Az ebben szereplő szavakat mi is megkérdezhetjük a géptől a *2. lista* programjának segítségével.

Amennyiben a kiegészítő információt szolgáltató szó vagy szavak azonosítása eredménnyel jár, akkor az EXOS 10 H rutin kiszámítja az EXOS változó memóriabeli címét és oda a megfelelő értéket beírja. Az azonosítás utáni memóriacím kiszámításának és beírásának interpreter általi megvalósítása a *3. listán* látható programmal lehetséges. A programból kiolvasható, hogy az EXOS változók helye a

BFC5H = 49093D címen kezdődik. Tehát a

SET BORDER 110

SET 27,110

POKE 49093 + 27,110 utasítás azonos hatást eredményez.

A gép mellé adott Felhasználói kézikönyv ismerteti a karakterek átdefiniálására szolgáló programot (93. oldal). Ennek a programnak a hatását is elérhetjük POKE utasítással.

Mielőtt beírnánk és futtatnánk a *4. lista* programját, egy-két megjegyzést vegyünk figyelembe. A SET CHARACTER CODE (123) utasítás az ä betű alakjának átírását teszi lehetővé. Minthogy a betűk alakját tartalmazó bittérkép a 255. szegmens 13440. helyétől kezdve található meg, csak a megfelelő címekre kell beírni a POKE utasítással a megfelelő értékeket.

A program segít a billentyűk átkódolásában.

A 260-as DATA sorba azt a kilenc értéket kell beírni, amit a karakter alakjának megtervezésekor kiszámítottunk. A program, elindítás után, kér egy billentyűlenyomást, és a lenyomott billentyűt a 210-240-es sorok segítségével átalakítja a DATA sor adatainak megfelelő alakra.

Egy szó, mint száz, még az olyan komfortos BASIC mellett sem kell elfeledkezni a POKE utasításról, annál kevésbé, mivel itt is vannak olyan feladatok, amelyek megoldására nincs más lehetőség.

SZ. LUKÁCS JÁNOS

MEGKÉRDEZTÜK AZ ENTERPRISE -RÓL

A Professional Szerviz 1985-ben alakult meg az Agroindustria Innovációs Vállalaton belül. Tizenkét vidéki városban az egész országot behálózó, egymással szoros kapcsolatban álló egységekből szervizhálózatot alakított ki. Az Enterprise gépek garanciális javítását kezdettől fogva ellátják. Az eközben gyűjtött tapasztalataikról kérdeztük meg Gottfried Tibor szervizvezetőt.

M. M. Miért a Professional Szervizre esett a választás, amikor szerződést kötött a forgalmazó az Enterprise gépek garanciális javítására?

G. T. A Professional a Novotrade márkaszervize, és ezért minket ajánlott a Centrumnak. Különben hagyománya van már nálunk a kisgépek javításának, mert a C Plus/4-et és egyéb iskola-számítógépeket is mi szervizeljük.

M. M. Nem vállaltak-e túlzottan nagy kockázatot azzal, hogy egy nálunk teljesen ismeretlen, új géptípussal kezdtek el foglalkozni?

G. T. Számunkra nem volt ismeretlen, ugyanis előzőleg kaptunk három gépet. Ezeket szétszedtük és megvizsgáltuk a szervizelés szempontjából. Azért sem volt annyira új, mert az Enterprise nagyon hasonlít a ZX-Spectrumra. Tulajdonképpen egy általános Z80 alapú gép, amelyet két koprocesszorral „okosabbá" tettek. Az üzlet anyagi kockázatát egy általunk felállított menedzsercsoport gazdaságossági mutatók alapján is jónak ítélte.

M. M. Elterjedt az a hír, hogy ezek a cég nyakán maradt, félig összeszerelt gépek voltak, és a megrendeléskor úgy „hányták össze" azokat.

G. T. Nagyon sok hír terjedt el, jó és rossz egyaránt. Ennek az lehet az oka, hogy soha nem látott mennyiségű gép jött be a magyar piacra, viszonylag alacsony áron. A számítógép olyan, mint a gépkocsi: mindenkinek a sajátja a legjobb. Én attól féltem, hogy az Enterprisezal is elkövetik azt a hibát, amit a C64-gyel: nem lesz hozzá hardver- és szoftverkiegészítés. Végül azonban gyorsan reagált a Novotrade, az „A" Stúdió és a Centrum is. Itt jegyzem meg, hogy minden hardverkiegészítőt is mi szervizelünk.

M. M. Van-e az Enterprise-nak tipushibája?

G. T. Ami van, az a fizikai kialakításából adódik, ugyanis a gép nagyon lapos. Az ebből eredő hátrányok persze „kivédhetők". Leggyakoribb a fóliatasztatúra törése, főleg a botkormány alatt. Nagyon sokszor elmondtuk, hogy a beépített botkormány nem játékra való. Előfordul, hogy a gép hátulján levő kis magnetofoncsatlakozó kitörik. A csatlakozókra vigyázni kell, mert elég gyengék. Nem tanácsos más magnetofont csatlakoztatni, mint amelyiket a géphez árulnak. Már csak azért sem, mert a magasabb remotáram elrontja a gépet. Igaz, tönkre nem teszi, de javítást igényel.

M. M. Nem hátrányos-e szervizelés szempontjából, hogy két típus érkezett be?

G. T. Nem! A két típus szervizelése megegyezik.

M. M. Egyik olvasónk megkérdezte, az Enterprise gépre is igaz-e, hogy a színes televízió színeit kiégeti?

G. T. Igaz, mert tévéműsor vételére is alkalmas! A számítógép a képcső azonos pontjait intenziven használja, és ezek a területek idővel bemattulnak. Ezt azonban több száz üzemóra idézi elő. Vigyázni is lehet a képcsőre. A szintelítettséget és a fényerőt le kell venni, és ha nem programozunk, akkor kapcsoljunk át másik csatornára. A játékprogramok nem ártanak, mert azok úgy viselkednek, mint a tévéadás.

M. M. A programok kazettáról általában jól tölthetők. Néhány olvasónk mégis panaszkodott, hogy bizonyos típusú tévék visszahatnak betöltéskor.

G. T. A Beijingnél és a legtöbb Videoton-tévénél mi is észleltük ezt a visszahatást.

M. M. Nekem ez elég elképzelhetetlennek tűnik. Hogyan fordulhat elő?

G. T. Valószínűleg egy földhurok alakul ki, de ennek megállapításához olyan műszerek kellenek, amelyek nekünk nincsenek a birtokunkban. Azt javaslom, hogy a két televíziót gyártó cég és a szerviz együtt vizsgálja meg ezt a dolgot. A Beijing akkor is lehetetlenné teszi a betöltést, ha a gép nincs rákötve, de a közelében működik. Sajnos a műszerek hiánya miatt nem tudok pontosabb választ adni. A Videotonnal a kapcsolatot már felvettük.

M. M. Hány Enterprise-t hoztak be javítani az elmúlt időben?

G. T. Az ügyfeleink száma szeptember elsejéig körülbelül 2400 volt. Többségük valamilyen tanácsot kért. A legtöbb kérdés a magnetofon beállításával volt kapcsolatban. Legtöbbször a fejen kellett állítani. A programbetöltés visszatérő gond, igaz, sokszor a hozzá nem értésből adódik.

M. M. Találtak-e konstrukciós hibát?

G. T. Tervezésit nem. Egyéb beállítási hibák előfordulnak, mint a szín-, illetve szinkronelterés.

M. M. Hogyan viseli az Enterprise a melegedést? Téved-e a gép hosszú üzem alatt?

G. T. Ilyen hibát nem tapasztaltunk. Ha a gép három órát hibamentesen üzemel, akkor nem téved. Ha rossz a gép, ez elég korán kiderül. Néhány jellemző hibalehetőség: a „tünet" attól függ, hogy melyik alkatrész melegszik. Ha a Z80 mikroprocesszor, akkor alapvető funkciókat sem kezel. Amennyiben a RAM-ban van a hiba, kisimul a kép és nem kezdi el a tesztet. A Nick chip hibája nem megengedett vízszintes csíkokat okoz, a Dave pedig eltorzuló kattogó hangokat eredményez, ha nem működik rendesen. Ha a kép „összetörik", az hangolással helyrehozható.

M. M. Ezek a gépek egyre idősebbek lesznek. Gyarapodnak-e ezáltal a meghibásodások?

G. T. Nem! Legalábbis egyelőre egyre kevesebb a meghibásodás.

M. M. Adna végezetül néhány tanácsot az Enterprise-tulajdonosoknak?

G. T. A meglévő programokat érdemes egy igen jó minőségű magnetofonnal rögzíteni. A gépet nem szabad zárt helyen, például íróasztalba beépítve üzemeltetni. Sok mérgelődéstől és költségtől megkímélhetjük magunkat, ha a tasztatúrát és a beépített botkormányt finoman kezeljük. Érdemes külön botkormányt vásárolni. Ez azért is fontos, mert a fóliatasztatúra-csere körülbelül 1500 forintba kerül, ha a gép már nem garanciális. A gépen lévő nem használt csatlakozók nyílt sebek. Ha ezek statikus elektromossággal feltöltött tárgyhoz érnek, az katasztrofális hibát okozhat. Ha nem használjuk a csatlakozót, ne vegyük le a jobb oldali takarólemezt. Végül egy kis kuriózum, amivel újjávarázsolható a Remot Reed relé. Az alábbi két programot futtassuk le:

```
10 TOGGLE REM1 10 TOGGLE REM2
20 GOTO 10      20 GOTO 10
```

A program ellenkező állapotba váltja a Reed relé állapotát, majd a 20-as sorban ciklusszerűen újra utasítást kap erre. A stop billentyű megnyomásáig egy halk, percegő hangot hallhatunk. A programot tíz-húsz másodpercig futtatva elvégezzük a relé mágnesezését.

PINKE GYÖRGY

FEDEZZÜK FEL EGYÜTT!

AZ ELLIPSZIS ÉS A TÖBBIEK

Nagyon izgalmas feladat és jó szórakozás egy „fekete doboz", az ismeretlen működését megismerni, leírni. Számítógépünket, az Enterprise-t is felfoghatjuk ilyennek, amiről azt tudhatjuk meg a Felhasználói kézikönyvből, hogy milyen BASIC nyelvű parancsokat, utasításokat fogad el és hatásukra mit csinál a számítógép. Akik a könyv segítségével már megismerték az Enterprise-t, azoknak csupán csak más utat ajánlunk, ami persze sok meglepetéssel szolgálhat.

Felfedező utunkra tehát minden Enterprise-tulajdonost meghívunk, figyelembe véve, hogy kezdők is vannak — és remélhetőleg egyre többen lesznek — közöttünk.

Cikksorozatunkkal csak a felfedezés irányítására vállalkozunk. Azt szeretnénk, ha a felfedezés örömét az olvasó élné át, akkor, amikor kis programjainkat futtatja, módosítja, kiegészíti és önállóan is készít programokat. Az utat olyan programok segítségével járjuk be, amelyek bemutatják a „fekete doboz" működését, alkalmazását, ugyanakkor igénylik az olvasó aktív közreműködését.

A Felhasználói kézikönyvre egyébként többször fogunk hivatkozni, mert abban megtalálható az egyes témakörök részletesebb leírása. A hivatkozás jelölése például: F/7, ami azt jelenti, hogy ezen az oldalon kezdődik a számítógép összeállítását leíró fejezet. Amennyiben mást írnánk a cikkben, mint ami a könyvben olvasható, úgy hívjuk segítségül a számítógépet a vita eldöntéséhez.

Segítségünkre lehet a tanulásban az alábbi algoritmus, amivel a javasolt tanulási módszert mutatjuk be.

1. Gépeljük be az első programot!
2. Indítsuk el a programot F1 START vagy a megszokott RUN parancscsal!
3. Figyeljük meg, hogy mi történik, és fogalmazzuk meg, hogy mit csinálnak a programban szereplő utasítások!
4. Olvassuk el a programhoz tartozó magyarázatot! Nézzük meg a segédkönyvekben az utasítások leírását!
5. Kísérletezzünk! Változtassuk meg a programban szereplő adatokat, és figyeljük meg, mi történik! Ellenőrizzük az értékhatárokat, az értelmezési tartományokat!
6. Oldjuk meg másképpen a feladatot! Egészítsük ki! Bővítsük a programot!
7. Keressünk feladatokat, amelyeknek a megoldásához felhasználhatjuk a tanultakat!
8. A cikk végére értünk?
   - Igen: próbáljuk meg hasznosítani a tanultakat! Jó munkát kívánunk! Vége.
   - Nem: összehasonlítjuk a következő programmal a már begépelt programot, és szükség szerint módosítjuk vagy töröljük a programot és begépeljük az újat, aztán folytatjuk a munkát a 2. ponttal.

No, de vágjunk neki az utunknak! Első lépésként gépeljük be az *1. listán* lévő programot. Rögtön egy kis segítség: rossz billentyű lenyomásakor a felesleges karakter(eke)t az ERASE vagy a DEL billentyű lenyomásával törölhetjük. Ha kihagytunk karakter(eke)t, akkor az INS billentyűvel csinálhatunk helyet. A kis botkormánnyal vihetjük a kurzort a javítás helyére. Az utasítássorok végén, illetve a javítás után ne feledkezzünk meg az ENTER billentyű lenyomásáról, mivel az utasítássorok beírása vagy a javítás csak ekkor válik véglegessé. Erről meggyőződhetünk, ha begépeljük a LIST parancsot. A parancsokat is az ENTER billentyű lenyomásával zárjuk le. Akkor is láthatjuk a program listáját, ha a 2-es funkcióbillentyűt nyomjuk le.

```
  1 REM  --- 1. program ---
 10 GRAPHICS HIRES 2
 20 PLOT 600,360
 30 PLOT ELLIPSE 200,100
 40 PLOT 600,360,PAINT
```

1. lista

```
  1 REM  --- 2. program ---
 10 GRAPHICS LORES 256
 20 PLOT 600,360
 30 PLOT ELLIPSE 200,100
 35 INPUT X
 40 PLOT X,360,PAINT
 50 GOTO 35
```

2. lista

```
  1 REM  --- 3. program ---
 10 GRAPHICS HIRES 2
 20 INPUT X,Y
 30 PLOT X,Y;
 40 GOTO 20
```

3. lista

```
  1 REM  --- 4. program ---
 10 GRAPHICS HIRES 256
 20 FOR I=1 TO 100
 30    LET X=RND(1280):LET Y=RND(620)
 40    SET INK RND(256)
 50    PLOT X,Y;
 60 NEXT
```

4. lista

```
100 REM  --- 5. program ---
110 GRAPHICS HIRES 256
120 SET PAPER 96
130 CLEAR GRAPHICS
140 LET X1=100:LET Y1=200
150 RANDOMIZE
160 FOR I=1 TO 200
170    LET X=RND(1280):LET Y=RND(720)
180    SET INK RND(256)
190    PLOT X1,Y1;X,Y
200    SET INK 96
210    PLOT X1,Y1;X,Y
220    LET X1=X:LET Y1=Y
230 NEXT
```

5. lista

A programot a RUN paranccsal indíthatjuk el. Miután megnéztük a képernyőn megjelenő ellipszist, értelmezzük a programban szereplő utasításokat!

10-es sor: kétszínű finom felbontású (HIRES) grafikus képernyőre rajzolunk.

20-as sor: a „ceruzát" a 600,360 koordinátájú képernyőpontra helyeztük. Ezzel jelöltük ki az ellipszis középpontját.

30-as sor: Rajzolunk egy ellipszist, amelyiknek 200, illetve 100 egység hosszúságúak a tengelyei. Egy egység két képernyőpont távolságának felel meg.

40-es sor: a „ceruzát" az ellipszis középpontjába helyezzük, és a PAINT utasítással be akarjuk festeni az ellipszist. A festés csak akkor indul el, ha a kiindulópont nem esik az ellipszis középpontját jelző pontra (F/82).

Egy grafikus pont több képernyőpontból áll. Változtassuk meg a 40-es sorban a koordinátákat és állapítsuk meg, hogy hány képernyőpontból áll egy pont a kétszínű grafikus képernyőn! Ezután a 10-es sor átírásával nézzük meg a többi grafikus képernyőt is. Ellenőrizzük az alábbi táblázatot, ahol

A: a különböző felbontású és színszámú grafikus képernyőt előállító utasításokat jelöli,
B: egy pontot előállító képernyőpontok száma vízszintesen,
C: függőlegesen,
D: a képpontok száma vízszintesen,
E: függőlegesen.

A	B	C	D	E
GRAPHICS HIRES 2	2	4	640	180
GRAPHICS LORES 2	4	4	320	180
GRAPHICS HIRES 4	4	4	320	180
GRAPHICS LORES 4	8	4	160	180
GRAPHICS HIRES 16	8	4	160	180
GRAPHICS LORES 16	16	4	80	180
GRAPHICS HIRES 256	16	4	80	180
GRAPHICS LORES 256	32	4	40	180

Gyorsabban megkereshetjük az értékeket a *2. listán* látható programmal. A 35-ös sorban az X változó értékét kérjük, azaz a festés kiindulópontjának vízszintes koordinátáját. Indítsuk el a programot, és a kérdőjel megjelenése után írjuk be a koordináta értékét, és nyomjuk le az ENTER billentyűt. Az 50-es sorról mindig a 35-ös sorra ugrunk vissza a GOTO utasítással, ennélfogva folyamatosan kéri a program a koordinátákat. Ebből a „végtelen ciklus"-ból a STOP billentyű lenyomásával léphetünk ki. Írjuk a 35-ös sorba az INPUT "PONT":X utasítást. Ekkor a PONT szó jelenik meg a képernyőn, és ez után kell begépelnünk a pont első koordinátáját. Így írathatjuk ki azt, hogy mit kérünk. Figyeljük meg, hogy csak a négy alsó sorban látszik a kiírás. Ennyi a szöveges képernyő, a képernyő többi része grafikus. Ide rajzolhatunk.

A *3. listának* a programjával szakaszokból ábrát rajzolhatunk a képernyőre. A szakaszok végpontjainak koordinátáit úgy kell megadnunk, hogy a két adat közé vesszőt teszünk és a második adat után lenyomjuk az ENTER billentyűt.

A program begépelése előtt adjuk ki a NEW parancsot, amivel töröljük az előző programot.

A 30-as sorban az PLOT utasításban a koordináták után a pontosvesszővel azt jelezzük, hogy letesszük a „ceruzát" a „papírra". Ez azt jelenti, hogy amikor a következő pontba mozgatjuk a „ceruzát", akkor rajzolni fog egy szakaszt.

Keretezzük be a képernyőt, azaz keressük meg a legkisebb és legnagyobb koordinátájú pontjait a képernyőnek!

Most a számítógép adja meg véletlenszerűen a szakaszok végpontjait *(4. lista)*. A begépeléshez segítség: az egyenlőségjel és a zárójelek begépelésekor a SHIFT billentyűt lenyomva kell tartani. A képernyőpontok vízszintes koordinátáit 1280-nál kisebb, függőleges koordinátáit 720-nál kisebb nem negatív egész számmal adjuk meg. Az RND-függvények ilyen számokat állítanak elő a programban ($0 \le X < 720$;$0 \le Y < 1280$). A 256 szín közül az RND-függvénnyel választjuk ki a „ceruza" színét (30-as sor).

A FOR és a NEXT utasítás együtt hoz létre egy ciklust, azaz ismétlődést. I értéke 1-től kezdődően egyesével növekszik 100-ig, tehát százszor ismétlődnek a FOR és NEXT utasítások között megadott utasítások, vagyis száz képernyőpont kerül a képernyőre (F/59).

- Írjuk a 15-ös sorba a RANDOMIZE utasítást!
- Töröljük az 50-es sorból a pontosvesszőt! Mi történik?
- Írjuk át az 50-es sort: 50 PLOT 640,360;X,Y.
- Adjuk ki a RENUMBER (SHIFT és a 3-as funkcióbillentyű), majd a LIST (2-es funkcióbillentyű) parancsot (F/46).

Most törölni fogjuk a megrajzolt szakaszt, csak egy szakaszt látunk majd a képernyőn *(5. lista)*. A kirajzolt szakasz végpontja lesz a következő szakasz kezdőpontja, ezért az X és Y változók értékeit áttesszük, és így megőrizzük az X1 és Y1 változókban (F/28).

A SET PAPER utasítással a papír színét állítjuk be, és szükség van a CLEAR GRAPHICS utasításra is, hogy a képernyő beszíneződjön. A vonalat úgy töröljük, hogy a „ceruza" színét a „papír" színére állítjuk (180-as sor) (F/84).

DUSZA ÁRPÁD

Mi a manó?

Fórum az Enterprise-ról

A Centrum Áruházak Vállalat és a Mikroszámítógép Magazin szervezésében fórum lesz az Enterprise géppel kapcsolatban. A feltett kérdésekre a Centrum, az „A" Stúdió, a Novotrade, a Professional Szerviz és az Enterprise cég NSZK-beli képviselői válaszolnak. A kérdéseket írásban a szerkesztőségünkbe várjuk, de a helyszínen is feltehetik. A fórum helyéről és időpontjáról a napilapokban jelenik meg tájékoztatás.

●

Enterprise-összeszerelés Magyarországon?

H. M. Windisch, az Enterprise Computer GmbH igazgatója elmondta, hogy magyarországi panelgyártást, illetve összeszerelést terveznek. Jelenleg tárgyalásokat folytatnak, de igen pozitívan ítélik meg a lehetőséget. Típusváltozást is terveznek az új Enterprise-gép standard DIN-szabványcsatlakozókkal is. A konstrukciót is modernizálták, a gép gyorsabb lesz, tárkapacitását bővítik, és önálló külső tasztatúra is csatlakoztatható hozzá. Így például az IBM PC-billentyűzet is. A jelenlegi elképzelések szerint az új gép olcsóbb lesz.

●

Enterprise a Szovjetunióban

Az Enterprise GmbH tárgyalásokat folytatott a Szovjetunióval. A várható igény 100 000 db/év. A Szovjetunióba szánt gépek nyomtatott áramkörű kártyáit egyelőre Magyarországon kívánják gyártatni. Ebben az ügyben is biztató tárgyalásokat folytatnak.

●

Enterprise PC

Az Enterprise cég PC-ket fog gyártatni, ezek kompatibilisek lesznek a jelenleg kapható gépekkel. Az új Enterprise PC a jövő év első negyedében jelenik meg az NSZK-piacon. Az Enterprise céggel megállapodtunk abban, hogy magazinunk részére folyamatosan kis híreket és anyagokat küld. Az általuk kiadott Enternews című kiadványból bármelyik cikket átvehetjük. A hozzánk érkezett olvasói kérdéseket is szívesen megválaszolják. Lapunk olvasói várhatóan kedvezményt kapnak az Enterprise cég müncheni irodájában.

Gaetsch Günterné rajza

Hardver

A sorozat alapgondolata — azon a régi felismerésen túl, hogy az elektronika és a számítástechnika elválaszthatatlan egymástól — a következő tapasztalatot summázza. A szoftver — a programok — jelentősége egyre nő, de az is tény, hogy az igazán jó (az adott számítógép nyújtotta lehetőségeket maximálisan kihasználó) programok megírásához a programozónak rendelkeznie kell alapfokú áramköri hardverismerettel is. Megerősíti ezt, hogy szaporodik az olyan berendezések, mikroprocesszort alkalmazó rendszerek száma, amelyek programvezérelten működnek. Az ilyen rendszerek tervezőinek és fejlesztőinek is szükségük van integrált hardver- és szoftverismeretekre.

Be- és kimeneti áramkörök

A mikroszámítógépes rendszerekben a külvilággal, tehát a környezettel a kapcsolatot a processzorhoz kapcsolódó be- és kimeneti egységek teszik lehetővé. Ezek olyan elemek, amelyeken keresztül a környezet felől bináris információkat juttathatunk a rendszerbe — ezek a bemenetek —, vagy pedig rajtuk keresztül bináris adatokat vihetünk át a környezetbe — ezek pedig a kimenetek. Szokták az angol elnevezésük alapján inputoknak (bemeneteknek) és outputoknak (kimeneteknek) vagy röviden I/O egységeknek is nevezni.

A mikroszámítógépben az adatokat a mikroprocesszor, a tár és a bemeneti áramkörök között párhuzamosan, az adatbuszon keresztül kezeljük. A külső adatokat azonban akár párhuzamosan — hasonlóan az adatbuszon való adatáramláshoz —, akár sorosan, vagyis bitenként, sorban egymás után kezelheti a mikrogép.

A mikrogépes rendszerek környezethez csatlakoztatásának leggyszerűbb eszközei az ún portok (I/O port). Alapvető sajátosságuk, hogy a hozzájuk érkező jeleket vezérlő parancsra mintavételezik, és mindaddig tárolják, amíg újabb mintavétel nincs. Ez a megoldás az időzítési problémákat jelentősen csökkenti. A belső adatbuszra kötött kimeneti port az adatkiviteli utasításkor mintát vesz a busz jeléből, és megőrzi akkor is, amikor az adatbuszon már egészen más információ van. A bemeneti portok egy része is képes a bemenetére kerülő információt megőrizni, és azt adott parancsra a belső adatbuszra kapuzni. Az egyszerűbb bemeneti portoknak nincs ilyen tárolási képességük, és a kapuzó jel hatására a bemenetükön lévő, éppen aktuális tartalmat juttatják a belső adatbuszra.

A következőkben először a párhuzamos adatátvitelt, majd a soros adatátvitelt tekintjük át.

Párhuzamos adatátvitel

A mikroszámítógépes rendszerek leggyakoribb adatátviteli formája. A mikroprocesszor oldaláról nincs lényeges különbség, hogy az adatforgalom a memóriával vagy a külvilággal kapcsolatos. Mivel a címbusszal címezhetünk, az adatátvitelre pedig az adatbuszt használjuk, a helyet csupán a vezérlőjelek segítségével különböztethetjük meg. A címbuszra adott címkombinációval választhatjuk ki az adatátvitel célját (angolul: destination, ejtsd desztinésön) vagy forrását (angolul: source, ejtsd szorsz). Az egység kiválasztását a címdekóder áramkör végzi, aminek a kimenete akkor aktív, ha a címsinen lévő címkombináció a dekóderen beállítottal megegyezik.

Az átvitel irányát az RD (olvasás), illetve a WR (írás) vezérlőjel határozza meg. Azt, hogy memóriával vagy a külvilággal kapcsolatos az adatátvitel, a MEMRQ (memory request, ejtsd memori rikveszt = memóriakérés) vagy az IORQ (I/O request = be/kimenet kérés) vezérlőjelek döntik el.

Ez a vázolt megoldás a Z80 processzorra jellemző. A 8080-as mikroprocesszornál a MEMRQ és IORQ jeleket egyetlen jellé vonták össze, ez az IO/M jel. A jel magas szintje be/kimeneti műveletet, alacsony szintje memóriaműveletet jelez. Bár a 16 bites címbuszon 65536 cím jelölhető ki, a 8 bites mikroprocesszorok tervezői csak a címbusz alsó felét használták perifériacímzésre. Így 256 perifériacím alakítható ki.

Konkrét példaként nézzünk egy nagyon egyszerű, Z80-as mikroprocesszorhoz kapcsolódó, 16 be- és 16 kimeneti vonalas perifériakialakítást, amely az *1. ábrán* látható.

Bemenetként olyan 8 bites meghajtó áramköröket (74LS244) használtunk, amelyeknek háromállapotú kimenetük van. Az áramkör a bemenetére kapcsolt bináris információt akkor tudja a mikroprocesszor adatbuszára és így a mikroprocesszorba küldeni, ha az engedélyező bemenetei (1G, 2G) aktívak, alacsony szintűek. Ilyenkor az áramkörnek az adatbuszra kapcsolódó háromállapotú kimenetei az eddigi nagy ellenállású (lebegő) állapotból aktív állapotba kerülnek, és a bemeneten lévő értékeket az adatbuszra kapuzzák.

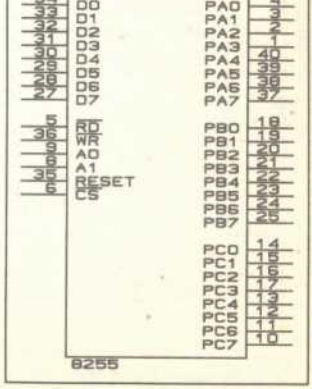
A be-, kimeneti kapuk címeinek dekódolását az U6A jelű, kettőről négyre dekódoló áramkör végzi. Ha az A8 = L és A1 = 1 szintű és IORQ = L, akkor U6A 4. lába lesz alacsony szintű, ami lehetővé teszi U1 és U3 kiválasztását. A cím felső hat bitjének értéke közömbös, azaz nem vesz részt a periféria címének kialakításában. Ezt a szokásos módon XXXXXX00 alakban jelezzük. Ez a kimeneti jel az RD és WR vezeték segítségével határozza meg azt, hogy az adott perifériacímről olvasni vagy a címre írni akarunk. Mivel tehát a 74LS244 áramkörnek két egymással ÉS kapcsolatban lévő engedélyező bemenete van, ezért egyszerűen a jelvezetékek rákötésével oldható meg a kapuk aktivizálása (a másik bemenetre az RD jelet kötjük). Hasonló módon az U2 és U4 periféria címe: XXXXXX01.

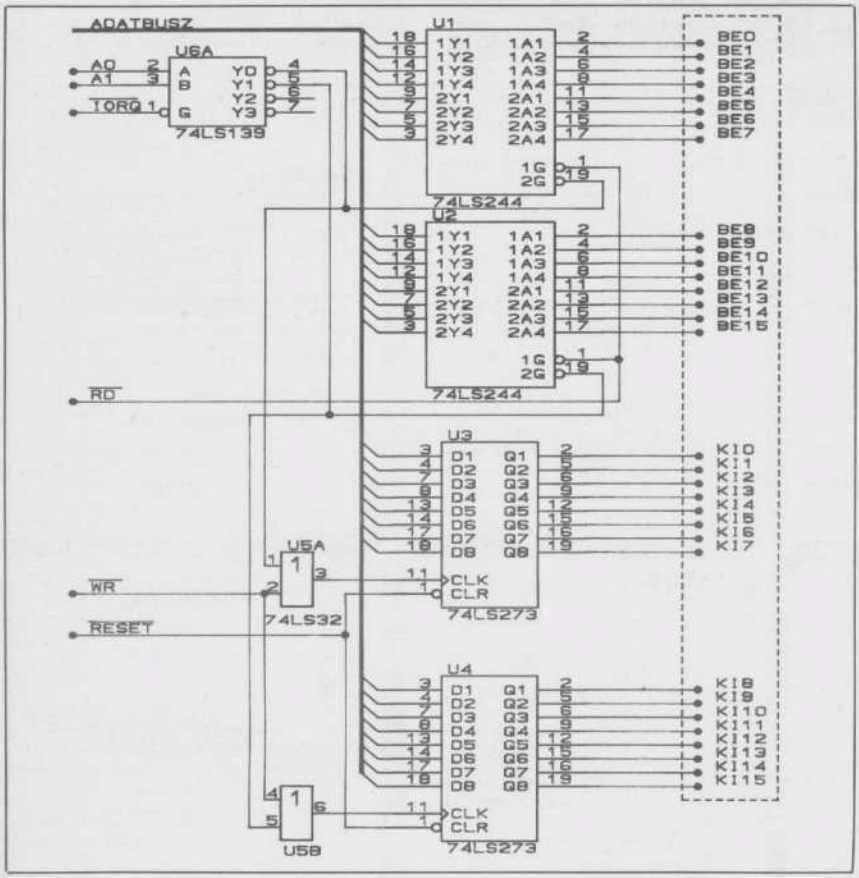
Kimenetként 8 bites 74LS273 típusú tároló áramköröket használunk. Ezek bemenetei az adatbuszra kapcsolódnak. A bemeneteken lévő tartalom akkor íródik be a tárolóba — és így a kimeneten is megjelenik —, ha a beíró bemenetén alacsony szint van. Ezt a jelet a címkiválasztó jel és a WR jel együtteséből állítjuk elő az U5 jelű VAGY kapukkal. Ezek kimenete akkor lesz a tárolók aktivizálásához szükséges L szintű, ha mindkét bemenetük L szintű. A RESET jel a tárolók alaphelyzetbe állítására (törlésére) szolgál.

Az előbbiek jól illusztrálják, hogy mennyire hasonló a mikroprocesszorok memória- és be/kimeneti egységének a kezelése. Ezek szerint lehetőség van arra, hogy egy memóriarész helyére be/kimeneti áramköröket tervezzünk, a kiválasztásukat az IORQ jel helyett az MREQ jellel végezzük el. Egy tárolási ciklus során az adatsín állapota beolvasódik a mikroprocesszorba, függetlenül attól, hogy honnan származik az adatsín tartalma. A memóriaként kezelt be/kimeneteket szokták memóriába ágyazott (memory mapped, ejtsd memori meppid) be/kimeneteknek nevezni. Ezek alapján nem meglepő, hogy van olyan mikroprocesszor, például a Motorola cég 6800-as sorozata — a család egyik eleme a C64 központi egysége —, amelynek egyáltalán nincs be/kimeneti utasítása! A működéshez szükséges be/kimeneteket memóriába ágyazva alakítják ki.

Be/kimeneti egységeknek — a szokásos TTL áramkörökön kívül — kifejlesztettek sokféleképpen felhasználható, a mikroprocesszor buszára közvetlenül kapcsolódó áramköröket is. Ezek tartalmazzák egyrészt azokat az áramköröket, amelyek a busszal teremtenek kapcsolatot, másrészt a külvilághoz kapcsolódó be/kimeneti áramköröket.

2. ÁBRA





1. ÁBRA

CS	Az IC kiválasztójele, aktív alacsony szintű
DØ—D7	Az adatbuszra kapcsolódó adatvonalak
RD	Aktív alacsony szintű jel, azt jelzi, hogy a CPU adatot olvas a 8255-ből
WR	Aktív alacsony szintű jel, azt jelzi, hogy a CPU adatot ír a 8255-be
RESET	Törlőjel, törli a belső regisztereket, a kapukat bemenet üzemmódba állítja, aktív magas szintű
PAØ—PA7	Az A adatkapu vonalai
PBØ—PB7	A B adatkapu vonalai
PCØ—PC7	A C adatkapu vonalai

A RESET jel megszűnésekor mindhárom kapu bemenetként működik. Ha így akarjuk használni, a tokot már programozni sem kell, azonnal használható. Ha más adatirányt vagy üzemmódot kívánunk elérni, akkor azt a használat előtt a parancsregiszterbe írt vezérlőszavakkal állíthatjuk be. A vezérlőszót jelentő bájtot az AØ = H és az A1 = H címbitek mellett írjuk a parancsregiszterbe.

A 8255 három üzemmódban használja kapuit:

MODE Ø üzemmód. Ilyenkor beállítható, hogy a négy csoport (A port, B port, C port alsó négy bitje, C port felső négy bitje) bemenetként vagy kimenetként működjön. A C portnak minden vonala a vezérlőszóval egyenként is be- vagy kimenetnek programozható. Az adatsin és a külvilág közötti kapcsolatot a vezérlőjelek alakítják ki, amit a *3. ábrán* foglaltunk ösze.

MODE 1 üzemmód. Ebben az üzemmódban a 8 bites adatátvitelű vonalakon kívül három vezérlőjel is közreműködik a megfelelő adatátvitelben. Az adatátvitel lehetséges az A vagy a B kapun keresztül, bemenetként vagy kimenetként működtetve. A kapuk vezérlése megszakításkérő logikával is kiegészül.

MODE 2 üzemmód. Csak az A kapu üzemelhet ilyen módon. Öt vezérlőjellel és megszakításkérő logikával segített, kétirányú adatátvitelt tesz lehetővé ez az üzemmód.

A tok részletes programozásáról bőséges magyar nyelvű irodalom található.

Egy feladatnál mindig el kell dönteni, hogy milyen típusú periféria-áramkörrel készítjük el az illesztést, ami függ a feladat jellegétől, a mikroprocesszor típusától, valamint az ár- és beszerzési viszonyoktól.

Az egyszerű felépítésű be/kimeneti portok és a nagy bonyolultságú periféria-áramkörök bő választéka lehetővé teszi csaknem minden illesztési feladat egyszerű és hatékony megvalósítását.

DR. KÓNYA LÁSZLÓ

3. ÁBRA

A1	AO	RD	WR	CS	MŰKÖDÉS
L	L	L	H	L	A kapu -> DO...D7
L	H	L	H	L	B kapu -> DO...D7
H	L	L	H	L	C kapu -> DO...D7
L	L	H	L	L	DO...D7 -> A kapu
L	H	H	L	L	DO...D7 -> B kapu
H	L	H	L	L	DO...D7 -> C kapu
H	H	H	L	L	DO...D7 -> par.reg.
X	X	X	X	H	az IC nincs kijelölve
H	H	L	H	L	ÉRVÉNYTELEN

Ezeknek az áramköröknek az üzemmódját — például azt, hogy egy adott vonal bemenetként vagy kimenetként működjön — az IC-ben lévő parancsregiszter(ek)be való írással határozhatjuk meg, amit úgy mondunk, hogy az áramkört programozzuk. Igen sokféle ilyen áramkör van. Jellegzetes példaként egyet, az Intel 8255 típusú, programozható perifériavezérlő egységet mutatjuk be. A tok bekötési rajza a *2. ábrán* látható.

Az áramkör 24 programozható be/kimeneti vonalat tartalmaz. A be- és kimeneti vezetékeket két, AC, valamint BC jelű csoportban kezeli az IC. Az AC csoporthoz tartozik a 8 bites A jelű adatkapu és a C kapu négy magasabb helyi értékű bitje, a többi a BC csoport.

Az áramkör jelcsatlakozásainak szerepe a következő:

AØ, A1	Az IC belső egységeinek kiválasztására szolgáló bemenetek

BÖRZE

TUTTI

ELECTROCOOP KISSZÖVETKEZET

— IBM PC kompatibilis gépek
— HARDVERTELEPÍTÉS SZERVIZ ÉS GARANCIA
— SZOFTVERES TÁMOGATÁS
— RÖVID HATÁRIDŐ

Cím: Bp., Üllői út 81. 1091
Tel.: 334-354

ECOSOFT
Számítástechnikai
Szolgáltató
Kisszövetkezet

IBM PC/AT
kompatibilis számítógép
ár: 229 000 Ft-tól
TURBO/32
32 bit, 24 MHz
ár: 479 000 Ft-tól.
Az általunk forgalmazott eszközök lízingelhetők is.
Tel.: 863-677

COBRA

ELEKTRONIKAI ÉS SZOLGÁLTATÓ KISSZÖVETKEZET
1097 Budapest, IX. Illatos út 7. Telefon: 476-160/388

KISSZÖVETKEZETEK!

Egyedülálló kínálatunk:

Számlakészítő program	19 900,— Ft
Számlanyilvántartó program	24 900,— Ft
Bér- és jövedelem-számfejtő program	24 900,— Ft
Főkönyvi könyvelőprogram	44 900,— Ft
	114 600,— Ft
COBRA—CONTO programcsomag	99 000,— Ft
IBM—XT kompatibilis számítógép	169 000,— Ft
STAR LC—10 nyomtató	49 000,— Ft
	317 000,— Ft

helyett mindezt már
299 000,— Ft-ért is megvásárolhatja!

MIKROPO BUDAPEST **IRODA:**
VI., Nagymező u. 51.
TEL.: 325-768

KARÁCSONYI MEGLEPETÉS:

— adatátviteli rendszerek
MIKROMOD, MODEM-CSALÁD
— CAD-rendszerek
nagy teljesítményű perifériák
magyar VERSACAD
— szuper mini számítógépek

ASY **Telefon: 415-166**
Kereskedelmi és szoftveriroda
1061 Bp., Liszt F. tér 10.
Telex: 22-4378

— **ASY—16 szupermikro számítógép** — 12 terminál
— VME busz
— UNIX
— **CRT TERMINÁLOK**
VT—52, QUT—102, Siemens 8160
— **BILLENTYŰZETEK**
— **MONITOROK**
— **INTEGRÁLT VÁLLALATI INFORMÁCIÓS RENDSZER**

UNIX környezetben üzemeltethető.

PERIFÉRIA

Elektronikai
Fejlesztő
és Szolgáltató
Kisszövetkezet
Bp. VII., Peterdy u. 30.
Telefon: 213-588

ajánlata:

— **P—XT:** 140 E Ft-tól + áfa
— **P—AT:** 200 E Ft-tól + áfa
— igény szerinti konfigurációk
— **FX—1000** PRINTER
90 E Ft + áfa

---

pc® **ez a védjegy a megbízható**

procontrol

termékeket jelöli

— Profi XT, AT, 386 gépek
— PLATON folyamatirányítók
— PORTAPRINT blokkolóórák
— T80 biztonsági rendszerek
— BCR vonalkód eszközök

PROCONTROL KISSZÖVETKEZET
6725 SZEGED,
VERESÁCS U. 28/B
TEL.: 62/21-165, 28-985,
TELEX: 82-726

ERIKA

a legújabb hordozható irodai elektronikus írógép.
CSÖNDES, PONTOS, MEGBÍZHATÓ,
ára: 25 200 Ft,
áfa-val együtt.
VÁSÁROLJA MEG nálunk:
Bp. VI., Népköztársaság útja 2.

ECONORG SZÁMÍTÁSTECHNIKAI KÜLKERESKEDELMI VÁLLALAT

1146 Bp., AJTÓSI DÜRER SOR 10.
Levélcím:
1393 Pf.: 319.
Telefon: 421-974
Telex: 22-6544

december havi kínálata:

a 80 386-os mikroproceszorra alapozott, multiuser üzemmódú

ACER SYS 32

rendszer, amely széleskörűen, változtatható kiépítésben, alap és alkalmazói szoftverekkel együtt kapható.

MINDENTUDÓ A „FEKETE DOBOZOKRÓL" II.

TRANZISZTORORSZÁGBAN

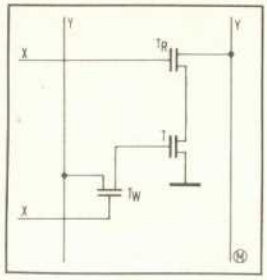
A tárak felépítésének elemzését a sztatikus MOS RAM-ok bemutatásával folytatjuk. A RAM betűszó arra utal, hogy a tár írható-olvasható, bár ha teljesen korrektek akarunk lenni, meg kell említenünk, hogy a RAM egyébként más fogalomnak, a véletlen hozzáférésű tárnak a rövidítése. A sztatikus tárolási tulajdonság azt jelenti, hogy a tár a benne lévő információt annak átírásáig vagy a tápfeszültség kikapcsolásáig megőrzi.

A MOS tranzisztorok előnyei a bipoláris tranzisztorokkal szemben (kisebb helyigény, kisebb táparamfelvétel, egyszerűbb gyártástechnológia) különösen jól kihasználhatók a nagy kapacitású tárak gyártásánál. Például a jelenleg már általánosan alkalmazott 64 k × 1 bites tár előállítása bipoláris technológiával elfogadható áron lehetetlen lenne.

A sztatikus MOS RAM-ok működésének megértéséhez vizsgáljuk meg az *5. ábrán* látható, 6 tranzisztorral felépített cellát! A tároló kétállapotú (bistabil), két inverterből (T_1—T_2, valamint T_3—T_4) áll. A T_5 és T_6 tranzisztoros kapcsolók teremtenek kapcsolatot a B és $\overline{B}$ vezetékkel; a kapcsolókat a W szóvezetékre adott jellel lehet zárni és megszakítani.

Állandósult állapotban, amikor írás-olvasás nincs, a W vezeték feszültsége a T_5 és a T_6 tranzisztorokat lezárja. Ilyenkor a kétállapotú tár megtartja eredeti állapotát.

Olvasáshoz a T_5 és a T_6 tranzisztorokat a W vezetékre kapcsolt jellel nyitni kell. Mivel a T_2 és a T_4 nyelőfeszültsége különböző,



6. ábra

a B és a $\overline{B}$ vezetékeken különböző áramok alakulnak ki. Ezeket a „vezetékek végén" lévő egy-egy olvasóerősítő érzékeli, így az erősítők kimenetén a tárolt információra jellemző jelek keletkeznek.

Íráskor a T_5 és a T_6 tranzisztorokat szintén nyitni kell. A tárolandó információnak megfelelő jeleket a kis impedanciájú kimenettel rendelkező íróerősítők a B és a $\overline{B}$ vezetékeken a bistabilhoz juttatják, rákényszerítve az új állapotot. Természetesen az olvasóerősítők ilyenkor áramot detektálnak, de — mivel írni és olvasni egyszerre értelmetlen — az olvasóerősítő kimeneti jeleit ilyenkor nem vesszük figyelembe.

A bemutatott cellából a már ismert módon mátrixot szervezhetünk. Egy cella kiválasztása a hozzá tartozó W és B—$\overline{B}$ vezetékek vezérlésével érhető el: az X vezetékeknek a W, az Y vezetékeknek a B—$\overline{B}$ feleltethető meg, vagy fordítva. A két író- és két olvasócsatlakozás megfelelő kapcsolástechnikával egyetlen ki-, illetve bemenetté egyesíthető.

A sztatikus tárolási módnak kétségtelenek az előnyei, de két hátránya is van. A tápáramfelvétel viszonylag nagy, ami egyébként CMOS technológiával csökkenthető; a bistabilt alkotó inverterek komplementer tranzisztorokból épülnek fel, de mivel a kettő közül az egyik zárva van, az inverter fogyasztása igen kicsi. A cella továbbá nagy helyet foglal a lapkán, ami nem teszi lehetővé egy tokban nagy tárkapacitás megvalósítását.

Állandósult állapotban a táparamfelvétel úgy csökkenthető, hogy a terhelőtranzisztorok (munkaellenállás-tranzisztorok) kapuelektródáit külön vezetékről táplálva, kisebb nyelőáramot állítunk be. Az eredeti áramokat csupán az írás és olvasás közben, tehát a működési idő kis százalékában kell biztosítani.

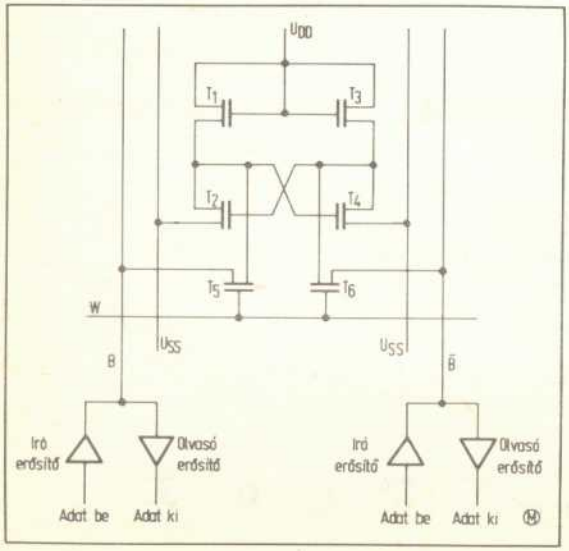
Állítsuk be a kapufeszültséget úgy, hogy az állandósult állapotban a terhelőtranzisztorok teljesen lezártak legyenek. Ekkor a tranzisztorok el is hagyhatók, és ezzel a megoldással tulajdonképpen már el is jutottunk az ún. dinamikus tárak szerkezeti felépítéséhez.

Dinamikus MOS RAM-ok

Dinamikus táraknak nevezzük azokat az eszközöket, amelyek a tárolt információt külső beavatkozás nélkül csak meghatározott időtartamig, néhány milliszekundumig képesek megőrizni. Hogy a tár tartalma el ne vesszen, az említett idő eltelte előtt a tár tartalmát ki kell olvasni és újra vissza kell írni. Ennek a folyamatnak a közismert elnevezése a frissítés. A teljesség kedvéért megjegyezzük, hogy ma már sok olyan típus van forgalomban, amely a frissítő áramköröket is tartalmazza, tehát a frissítésről nem a felhasználónak kell gondoskodnia. Neki az eszköz sztatikus tárként látható, erre utal elnevezése is: kvázisztatikus RAM.

Dinamikus tárcellaként általában kondenzátorokat alkalmaznak. A megoldás a MOS táraknál kézenfekvő: a kondenzátorok az áramkörben megtalálhatók (a C_{GS}, esetleg a C_{GD} tranzisztor-elektróda kapacitások), és a kisütési időállandók a MOS tranzisztorok nagy ellenállásértékei miatt — a kis kapacitások ellenére — éppen elégg nagyok.

A dinamikus tárcella létrehozásához és működésének vizsgálatához induljunk ki az 5. ábrán látható sztatikus tárcellából! Ha nem kívánjuk meg a sztatikus tárolási tulajdonságot, a T_1 és a T_3 terhelőtranzisztorok elhagyhatók. Mivel a T_2 és a T_4 — a bistabil multivibrátor elvéből következően — ellenkező állapotban vannak — az egyik nyitott, a másik zárt —, az egyik az infor-



5. ábra

máció elvesztésének veszélye nélkül el is hagyható. Ekképpen háromtranzisztoros áramkörhöz jutunk, amelyben egy-egy tranzisztor látja el a tárolást, a tárolótranzisztorba való beírást és a tárolótranzisztoroknak az összekapcsolását az olvasóvezetékkel (ezeknek a jelölése a *6. ábrán* rendre: T, T_W, T_R). A létrehozott áramkörök különféleképpen szervezhetők mátrixba a cellakijelölés és az írás-olvasás (frissítés) szempontjából.

A cellát a mátrixba 2 darab X és 2 darab Y vezető köti be. A vezetékek szerepének megértését a *7. ábra* segíti.

Íráskor az „Írásengedélyezés" vezetékre adott impulzus kinyitja a T_W tranzisztort, így a C_{GS} tárolókapacitás az „Íróvezeték"-en keresztül a tárolandó információnak megfelelően feltöltődhet.

Olvasás előtt az „Olvasóvezeték"-et rövid időre pozitív feszültségre kell kapcsolni, aminek hatására a C_R vezetékkapacitás feltöltődik. Ezután az „Olvasásengedélyezés" vezetékre kapcsolt impulzussal a T_R tranzisztort vezetővé tesszük; ekkor a T tárolótranzisztor és az olvasóvezeték között létrejön a kapcsolat. Ha a C_{GS} kapacitás töltése révén a kapu pozitív feszültségű, a T tárolótranzisztor vezetni fog, és az olvasóvezeték C_R kapacitása rajta keresztül kisül. Ha a kapufeszültség 0 V körüli, T zárva van, és az olvasóvezeték kapacitása eredeti állapotában (pozitívra töltve) marad. Az olvasóvezeték „végén" lévő olvasóerősítő — mintegy „megmérve" a vezeték feszültségét — állítja elő a tárolt információra jellemző kimeneti jelet.

A három tranzisztorból felépített tárolócella tovább egyszerűsíthető, ha a T_W és a T_R tranzisztor helyett egyetlen tranzisztort (T_G) alkalmazunk az írás és az olvasás engedélyezésére, továbbá elhagyjuk a T tárolótranzisztort is, és a tárolást egy kondenzátorral oldjuk meg. Az írás és olvasás egyetlen tranzisztorral való vezérlése egyben a vezetékek számának csökkentését is eredményezi; egy cella kiválasztására egy-egy X és Y vezeték lesz elegendő. A cella felépítését és kapcsolatát a vezetékekkel a *8. ábra* szemlélteti.

Beíráshoz a „Szókijelölés" (X) vezetéket kell vezérelni. Ekkor a T_G tranzisztor kinyit, és a C_T tárolókapacitás a „Bitkijelölés" vezetéken lévő feszültségnek — ezt az íróerősítő szolgáltatja — megfelelően töltődik fel.

Az olvasás előtt a C_R bitvezeték-kapacitást pozitív feszültségre kell feltölteni. Ekkor a K_1 zárt és a K_2 nyitott állapotban van. A pozitív feszültség akkora legyen, hogy az I_2 inverter munkapontját az aktív szakasz közepére, az U_{be}—U_{ki} jelleggörbe állapotváltozást leíró „meredek" részére állítsa be.

Ezután K_1-et nyitjuk és K_2-t zárjuk, azaz a két invertert flip-floppá kapcsoljuk össze. Ha most a T_G tranzisztort a szókijelölő vezetékre adott jellel kinyitjuk, a C_T tárolókondenzátor — töltöttségi állapotának megfelelően — megváltoztatja a C_R bitvezeték-kapacitás feszültségét, és így a flip-flopot valamelyik határozott állapotába billenti. A cella a háromtranzisztorosnál lényegesen egyszerűbb felépítésű, de amig annál a kiolvasás a tárolt információt nem törli, az egytranzisztoros elemnél a kiolvasás törlődő.

Az ismertetett tárolócella n-csatornás szilíciumkapu technológiájú kivitelben a jelenleg már közhasználatúnak tekinthető közepes-nagy kapacitású dinamikus RAM-ok alapeleme (16 k × 1 bit, 16 k × 4 bit, 64 k × 1 bit). Egyszerű felépítése miatt bonyolultabbak a kiszolgáló áramkörei — például egy 16 k × 1 bites RAM-ban 128 × 128-as mátrixot és így 128 szókijelölő áramkört, továbbá 128-128 író- és olvasóerősítőt kell alkalmazni —, de a ma már elérhető nagy alkatrészsűrűség lehetővé tette, hogy ezeket a tárolóchipen helyezzék el. A jelenleg forgalomban lévő áramkörök binárisan címezhetők, a hozzáférési idő 100-200 ns körül van, a frissítéshez szükséges egységeket részben vagy egészben tartalmazzák és TTL kompatibilisek.

A fixtárakról általában

A fixtárak olyan eszközök, amelyeknek tartalma csak olvasható, illetve a típusok egy részénél megváltoztatható ugyan, de nem üzemszerűen és csak hosszabb idő alatt. (Az angol Read Only Memory — csak olvasható tár — kifejezésből kialakult ROM betűszó a szakmában általánosan használt megnevezésük.) Kétféle típusuk az újra nem programozható és az újraprogramozható ROM.

Az újra nem programozható tárak információtartalma kétféle módon hozható létre. Az egyik megoldásnál a tartalmat a gyártáskor helyezik a tárba. Ez azt igényli, hogy a felhasználók a szükséges információt a gyártó cégnek átadják, aminek alapján a „készregyártás" elvégezhető. Mivel a felhasználók az információt általában nem szívesen adják ki, továbbá az említett eljárás hosszadalmas és csak nagy sorozatú gyártáskor kifizetődő, többnyire a beírás másik formáját használják, amikor is a felhasználó írja be az információt az „üres" tárba.

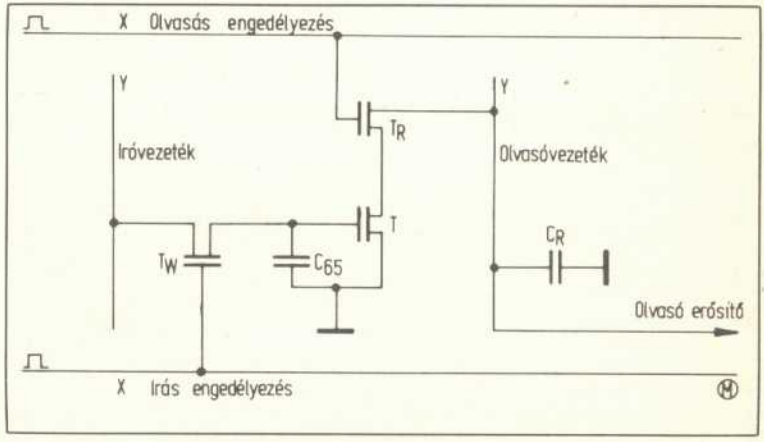
Az újraprogramozható tárak programozását általában a felhasználók készítik el saját igényeik szerint. A tár csak olvasható ugyan, de megfelelő — elég hosszadalmas — eljárással tartalma megváltoztatható. A régi tartalmat ultraibolya vagy röntgensugárzással 10-30 perc alatt lehet törölni, az új tartalmat beírni a módszertől és a tár kapacitásától függően néhány perc alatt.

A hardvertechnikában jártasabbaknak említésre méltó, hogy a ROM-okat nem csak tárolásra (például egy BASIC interpreter „beépítése" egy HC-be), hanem annál sokkal szélesebb körben lehet alkalmazni. Ha arra gondolunk, hogy egy 8 címbit bemenetes és 8 tartalombit kimenetes ROM egyes rekeszeinek címzésénél — bármikor is végezzük ezt — mindig ugyanaz az információ jelenik meg a kimeneteken, világossá válik, hogy ez az áramkör nem más, mint egy olyan kombinációs logikai hálózat, amelynek 8 bemenete és 8 kimenete van, azaz nyolc 8 változós függvényt valósít meg.

MOS ROM-ok

Az újra nem programozható MOS ROM-ok cellája igen egyszerű: egy MOS tranzisz-

7. ábra

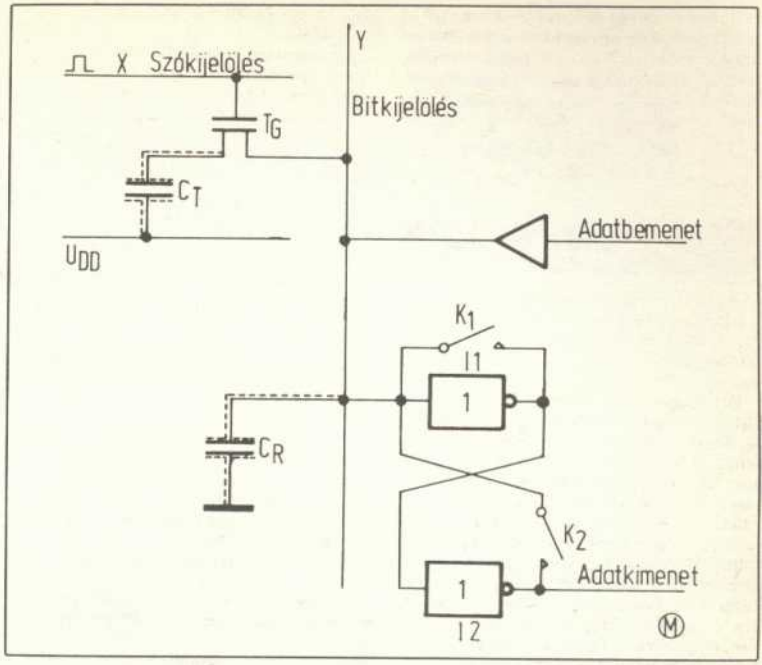


tor, amely az információbit értékétől függően vagy létrehoz vagy nem hoz létre kapcsolatot a szó- és olvasóvezeték között. Egy négy cellát tartalmazó mátrixrészlet a *9. ábrán* látható.

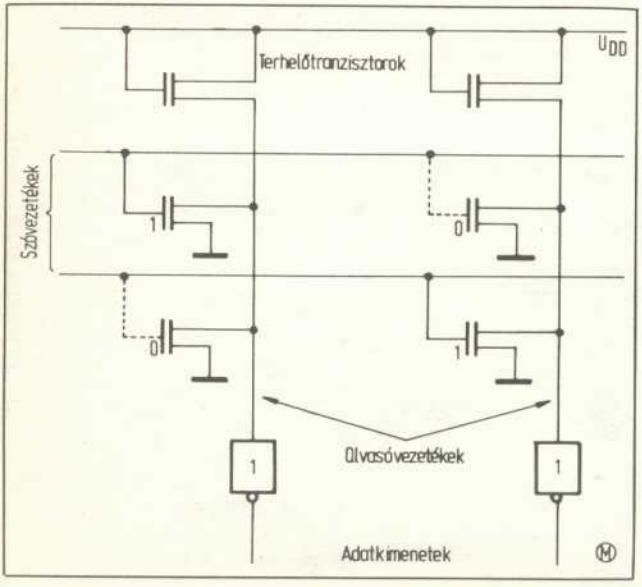
Azokon a helyeken, ahol a „Szóvezeték"-hez a kapu csatlakozik, a cella 1-et tárol, a többi helyen (az ábrán a szaggatott kapubekötés) a tárolt bit értéke 0. A „Szóvezeték" vezérlésekor a 1-et tároló tranzisztorok kinyitnak, és az „Olvasóvezeték" potenciálját — szakzsargonnal kifejezve — lehúzzák. Az adatkimeneteken az invertálás révén H szint jelenik meg, az 1-es kiolvasását jelezve. A cellák létrehozása technológiailag egyszerű: a 0-át tároló cellák tranzisztorainak kapu-oxidrétegét lényegesen vastagabbra készítik, így a kapufeszültség a nyelőforrás-csatornát nem képes vezetővé tenni, nyelőáram tehát ezeken a tranzisztorokon nem folyik. A jelenség egyszerűen kifejezve olyan, mintha a tranzisztor ott sem lenne.

Az újraprogramozható tárak megvalósításához általában a FAMOS tranzisztort (Floating gate Avalanche injection MOS — lebegő vezérlőelektródájú lavinafeltöltésű MOS) alkalmazzák. A FAMOS kapuelektródájának kivezetése nincs, körös-körül nagy tisztaságú, tehát nagy ellenállású sziliciumoxid veszi körül, amelynek vastagsága az n-típusú csatorna felé mintegy 100 nm. A kapu tehát meghatározatlan feszültségen van, „lebeg".

Ha nagy negatív (−40 . . . −50 V) feszültséget kapcsolunk a nyelő és a kapu közé, akkor e feszültség nagy része a lezárt



8. ábra



9. ábra

kapu-nyelő p-n átmenetre esik. Az átmenetnél lavinaeffektus jön létre, és az elektronok a vékony sziliciumoxid rétegen keresztül néhány ms alatt feltöltik a kaput. Ez a töltés elegendő ahhoz, hogy a kapuelektródán −10 . . . −15 V küszöbfeszültséget hozzon létre; ugyanakkor azokon a helyeken, ahol nem lépett fel a lavinaeffektus, a küszöbfeszültség max. −1 . . . −1,2 V lesz. Az így „programozott" tranzisztorok különbözőképpen fognak viselkedni az áramkörben: az egyik nyitott (vezető), a másik zárt lesz. A minden oldalról szigetelt kapu töltése igen hosszú ideig megmarad: a kisütési áram 10^{-40} A nagyságrend körüli! A küszöbfeszültség 10 százalékos megváltozásához szükséges időt közelítő számításokkal 80-100 évre becsülik.

Az információ törléséhez sugárzást kell alkalmazni. Ha a tokon van kvarcüveg ablak, ultraibolya fény használható, ha nincs, a törlés röntgensugárzással végezhető el. A szükséges sugárzási intenzitást és időt a gyártó cégek megadják.

Az egy tokban megvalósítható MOS ROM-ok kapacitása jelenleg 64—256 kbit, a hozzáférési idő 100-200 ns.

NAGY IMRE

T O R O R S Z Á G B A N

Sorozatunkban azokat az új hardver- és szoftvertermékeket ismertetjük, amelyek várhatóan általánosan elterjednek, és meghatározó szerepük lesz a fejlődés irányainak kialakításában.

Merre tart a világ?

MONITOROK

A monitorgyártásban ugrásszerű változás nem következett be. A képcsövek laposodnak, felbontásuk nő, és egyre gazdagabb a színválasztékuk is.

SONY

A GDM—1953 19" képméretű, nagy felbontású, CAD/CAM célra tervezett monitor. Clem Shemanski, a cég eladási igazgatója szerint az általuk gyártott és ebben a monitorban is használt Trinitron® cső miatt a pontok élessége és fényessége jobb, mint a hasonló felbontású más termékeké. Előnye még a készüléknek, hogy lapos a képcsöve, a képszélek torzításmentesek, ugyanúgy torzításmentes a rajzvisszaadási képessége, valamint a fényvisszaverődést is csökkentették. Az *1. képen* látható monitor adatait az *1. táblázat* tartalmazza.

A CPD—1303 Multiscan monitor *(2. kép, 2. táblázat)* a legnagyobb felbontású a hasonló méretűek között. Ebbe szintén Trinitron® csövet építettek. Az IBM PC/XT és PS/2 sorozatához, valamint az Apple Macintosh II tipushoz ajánlják. A különlegesen nagy felbontóképessége miatt, a cég igazgatójának véleménye szerint, nemcsak a mai színes monitorokkal szemben támasztott követelményeknek felel meg, hanem a jövőben is kielégíti ezeket. Eleget tesz a CGA módtól a VGA módig az összes elterjedt grafikus felbontási előírásnak. Ráadásul szinte tökéletesen, ami nem mondható el a többi gyártmányról, mert van, amelyik az egyik, van, amelyik a másik szabványhoz idomul jól, míg másokhoz kevésbé. A bemenőjel egyaránt lehet analóg RGB vagy digitális jel. Az előző monitornál felsorolt előnyök, az azonos típusú képcső használata miatt, a CPD—1303 Multiscant is jellemzik.

HITACHI

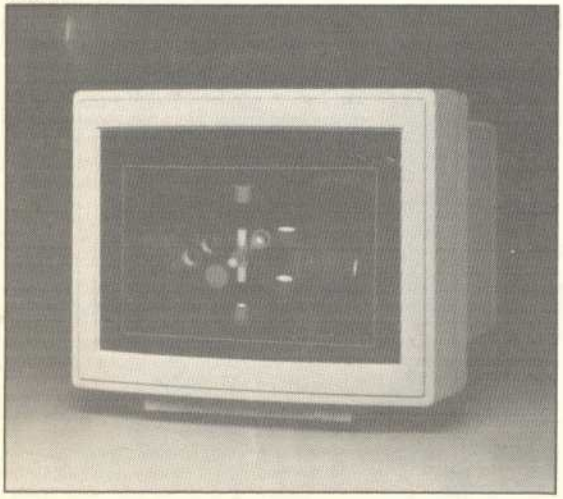
A cég termékeivel a számítástechnikai hardvergyártás majdnem teljes választékát kínálja, és ezek egyaránt ismertek, akár integrált áramkörről — például a 68 számkezdésű mikroprocesszorokról, statikus és dinamikus RAM-okról, a különféle ROM-okról stb. —, akár részegységről — különféle háttértárolókról — vagy önálló berendezésekről — plotterekről, rajzdigitalizálókról — legyen szó. Szinte természetes, hogy a monitorgyártás is „erőssége" a Hitachinak.

A HM- (High Tech Monitor) sorozatnál, ahogy az előző cégnél is, néhány lényeges és azonos megoldást alkalmaztak. Ilyen a DDC (Dynamic Digital Convergence modul) és az Inline-tube. A DDC egy beépített célszámítógéppel teszi lehetővé a konvergencia beszabályozását az összes előforduló frekvencián. A számítógépes irányítás gondoskodik egyrészt arról, hogy a beállított értékhez rendelje a többi paramétert, másrészt ezeket az értékeket tartja. Ezzel a megoldással mintegy megkétszerezték a felbontóképességet. Az Inline-tube nagyobb fényerőt ad, ezáltal lehetővé válik az energiatakarékosabb megoldás. A kábelezése különleges, emiatt a monitort dönteni, forgatni lehet anélkül, hogy az ilyen használat a kábeleknél problémát okozna.

A sorozat rendkívül nagy képméretű tagja a 4625 *(3. kép)*, amelynek mérete 25". Az 5219 sávszélessége egészen nagy, a 4219 pontmérete pedig igen kicsi. A 4619 típus nagyon hasonlít az előzőre, csak a pontmérete nagyobb, viszont a bemenőjelének szintje alacsonyabb. A négy típus adatait a *3. táblázat* foglalja össze.

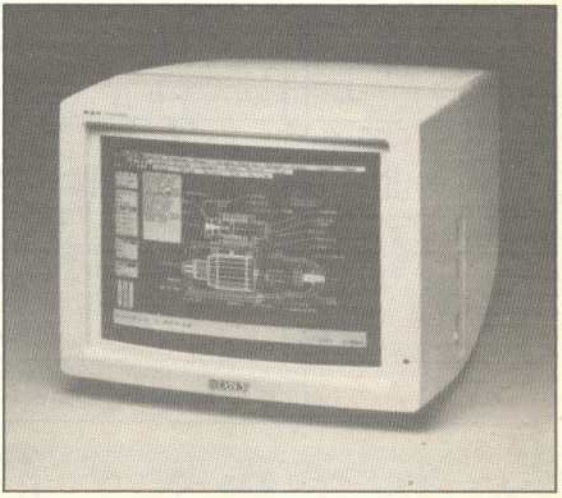
— SE —

1. kép



SONY® GDM-1953 Monitor

2. kép



SONY® CPD-1303 Multiscan Monitor

3. kép

1. táblázat

Megnevezés	Érték
Képterület	343× 274 mm
Pontméret	0,31 mm
Felbontás	1280(H)× 1024(V) pont
Horizontális szinkron	64
Sávszélesség	100 MHz
Vertikális szinkron	60 Hz
Befoglaló méret	480× 446× 535 mm
Tápadatok	100—120 V, 50/60 Hz, 2,6 A max.

2. táblázat

Megnevezés	Érték
Képterület	274× 207 mm
Pontméret	0,26 mm
Felbontás	1024(H)× 768(V)
Horizontális frekvenciasáv	15,75—36 kHz
Vertikális szinkron	50—100 Hz
Befoglaló méret	370× 330× 415 mm

3. táblázat

Megnevezés	Érték			
	4219	4619	4625	5219
Képterület	340× 270	340× 270	450× 355	341× 273
Pontméret	0,31/0,26	0,31	0,37	0,26
Konvergencia	0,1/0,3	0,3/0,5	0,5	0,3/0,5
Felbontás	1280× 1024	1280× 1024	1280× 1024	1600× 1280
Horizontális frekvenciasáv	61—65	61—65	62—65	78—90
Vertikális frekvenciasáv	55—65	55—65	55—65	60—75
Horizontális visszafutási idő	3,5 µs	3,5	4,0	3,2
Vertikális visszafutási idő	550 µs	550	600	530
Sávszélesség	15—100	15—100	15—100	15—100
Névleges esési idő	5 ns	5	5	3
Bemenőjel	analóg	analóg	analóg	analóg
Bemenőjelszint	0,7—1,4 V	0,5—1,5	0,7—1,4	0,7—1,2
Külső szinkronszint	TTL	4,0	4,0	2,5
Külső szinkronszint szabvány	RS343A	RS343A	RS343A	RS343A
Tápadatok	87—132 V, 2 A	175—264 V, 3,7	47—63 Hz 5	2,4

ZX–Spectrum

Segíts magadon!

Ötletek a képernyőtartalom kinyomtatására

Elég régóta használom a Masterfile programot. Igen praktikusnak találom, hogy az adatokat bármely szempontból képes rendezni. Ám sokszor nehézségem támadt amiatt, hogy az ilyen módon rendezett adatokat nem láthattam, elemezhettem nyomtatásban, mert nincs Spectrum nyomtatóm. Nagyon megörültem, amikor lehetőségem nyílt egy Brother M—1109 típusú nyomtató időnkénti használatára. Sajnos azonban a nyomtató „nem érti" a COPY parancsot. Először egy BASIC program segítségével nyomtattam, de ennek a megoldásnak két hibája volt. Az egyik, hogy egy képernyő kinyomtatása hozzávetőlegesen tíz percig tartott, a másik ennél sokkal alapvetőbb. De előbb nézzük a képernyőt!

Mindazoknak, akiknek valamilyen RS232 csatlakozású nyomtatójuk van a Spectrumhoz, gondot jelent a képernyőtartalom kinyomtatása. Ennek oka a számítógép képernyőjének a szervezése. A gépkönyv szerint a képernyő két részből áll:

— képtartalomból (Display File)
kezdőcím: 16384 (4000H)
— színtartalomból (Attributes)
kezdőcím: 22528 (5B00H)

Ez azonban a mi szempontunkból igen kevés információ. Nézzük a részleteket! Mindenki, aki már betöltött képernyőtartalmat a képernyő-memóriába, észrevette, hogy a képernyő három (nyolc betűsoros) részre van osztva.

	Kezdőcím		Utolsó cím		Képernyősor
	DEC	HEX	DEC	HEX	
1. harmad	16384	4000	18431	47FF	0— 7
2. harmad	18432	4800	20479	4FFF	8—15
3. harmad	20480	5000	22527	57FF	16—23

Milyen az első karakter a memóriában? Mely memóriacimek és milyen formában tartalmazzák?

	128	64	32	16	4	2	1
16384	.	.	.	.	.	.	.
16640	.	.	.	.	.	.	.
16896	.	.	.	.	.	.	.
17152	.	.	.	.	.	.	.
17408	.	.	.	.	.	.	.
17664	.	.	.	.	.	.	.
17920	.	.	.	.	.	.	.
18176	.	.	.	.	.	.	.

Ez azt jelenti, hogy a rajzolat vízszintes vonalkákból alakul ki. Ha a képernyő koordinátahálózatát nézzük, a 16384-es memóriarekesz bal szélső bitje a (0,175) pontnak felel meg.

Mélyedjünk bele a kettes számrendszer és a Spectrum képernyőszervezésének rejtelmeibe. A Spectrum egy memóriarekesze alkalmas egy 0—255 közötti szám tárolására. Hogy miért? A memóriarekesz nyolc bitből áll, és egy bit ugyebár vagy 0 vagy 1 lehet. Tehát a 00000000 = 0 és az 11111111 = 255. Egy betűhely, amit a program írásánál egy egységnek vettem — majd kiderül, miért —, nyolc memóriarekesz tartalma, tehát 8 × 8 bit. Visszatérve a kettes számrendszerre: a 0 és 1 számok a 2 hatványainak megfelelően helyiértékkel rendelkeznek, mint ahogy az alábbi ábrán látható:

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

BASIC utasítások segítségével a képernyő pontjai — azaz a képernyőrekeszek bitjei — egyenként címezhetők. A PLOT utasítással a biteket 1-be állíthatjuk, vagyis rajzolhatunk, a POINT utasítás segítségével pedig kiolvashatjuk a képernyőpontnak, tehát a memóriarekesz egy bitjének az állapotát. A képernyő alsó két sora ilyen módon nem címezhető, mert a képernyő (0,0) pontja alulról a harmadik sor bal alsó sarka, és a POINT, valamint a PLOT utasítás nem ismeri a negatív számokat. Ezért BASIC-ből a 22. és 23. „hibaüzenet" sorok pontjai nem címezhetők. Vagyis ez a BASIC másik hibája.

A nyomtatók típustól függően nyolc vagy kilenc függőlegesen elhelyezett tűvel írnak. Az a nyomtató, amivel dolgozom, bitnyomtató üzemmódban nyolc tűvel nyomtat, ezért a program erre készült. Természetesen nehezíti a dolgot a képernyő harmadokra való osztása is. Nagyobb gond azonban a bitek „átfordítása", vagyis ahogy az előbb leírtam: a vízszintes vonalakból felépített képet függőleges vonalakra „átírni". Hogyan oldottam meg végül a problémát?

Míg a Spectrum memóriájában a jobb szélső pont értéke 1 és a bal szélső ponté 128, a nyomtatóban a legalsó pont az 1 és a legfelső a 128. Tehát nem elég a memóriarekeszek egyszerű PEEK-elése, hanem a kapott számot át kell számolni a kettes számrendszerbe, ami elképzelhetetlenül lassú lenne. Ezért a pontokat grafikusan kell egyenként kiolvasni a POINT utasítással. Emiatt lassú a BASIC program.

Térjünk vissza a képernyőhöz! Egy karakter elrendezése feljebb látható. A 18176-os memóriarekeszt a karakter báziscímének neveztem el. A képernyő három harmadra van felosztva. Egy harmad nyolc sor, egy sorban 32 karakter van. Azaz egy harmad 256 betűhely. Ezeknek a báziscímei sorban egymást követő számok. Tehát

az első harmad:
18176, 18178, 18179, . . . , 18431
a második harmad:
20224, 20225, ... , 20479
a harmadik harmad:
22272, 22273, ... , 22527

Az első képernyőcímnél (ami az első betű teteje) a báziscím 7 × 256 = 1792-vel nagyobb. A harmad első báziscíménél az utolsó 255-tel nagyobb (ennyi betű van a harmadban).

Nézzük tehát a program lényegét!

A program változói:

IY > CIM a betűhelyen levő nyolc memóriarekeszt számláló rekesz címe
IY+1 > egy memóriarekesz 8 bitjét számolja
IY+2 > a sor 32 karakterét számolja
IY+3 > a képernyőharmad 256 karakterét számolja
IX > PUFF kinyomtatandó memóriarész címe
HL > a betűhely báziscíme

A fontosabb szubrutinok:

ESCBIT a nyomtató beállítása minden képernyősor előtt
START a program kezdete
SOROK az (IY+2) és az (IY+3) beállítása
HARM egy harmad kiolvasása és kiírása
VALT a képernyőharmad váltáskor aktiválódik
VISSZ a BIT rutinból való visszatérés a főprogramba
PRINT egy betűhely kiíratása
BIT egy betűhely „átfordítása"
VEGE kilépés a programból

A szubrutinok címeit lásd a táblázatban!

A program csak relatív címeket használ, ezért bárhová betölthető. Hossza 240 bájt. A program elkészítésekor az vezérelt, hogy 256

BET	5B80	BETU	5B84
BIT	5BB2	CIKL	5B98
CIM	5BF0	ESCBIT	5B1D
FOLYT	5B61	HAR	5B54
HARM	5B4E	LEP	5BC0
LEPES	5BC8	OSZL	5BBC
PLUSZ	5B7A	PRINT	5B92
PUFF	5BE8	SOROK	5B46
START	5B3E	TOV	5BB0
TOVA	5BCF	TOVAB	5BCC
VALT	5B6D	VEGE	5BE6
VISSZ	5B89		

bájtnál rövidebb legyen, hogy ha a memóriába máshová nem helyezhető el, akkor a PRINTER PUFFER-ben elférjen. Ezért is betűnkénti és nem soronkénti a kiíratás.

TABULÁCIÓ: a POKE (betöltési cím + 12), N segítségével.

A KIÍRATÁS MÓDJA: POKE (betöltési cím + 41), M ahol $0 <= M <= 7$.

PROGRAMBÓL VALÓ KILÉPÉS: SYMBOL SHIFT + bármelyik billentyű lenyomására (kivétel a CAPS SHIFT).

A program leírása:

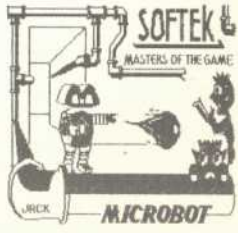
60—70	csatornamegnyitás
80—150	TAB-beállítás
120	TAB mértéke
160—190	ESC 1: sorköz nincs
200	ESCBIT rutin végén H-tól függő ugrás van, a program során H-ban nem 0 van
210—240	kocsi vissza, soremelés
250—260	TAB-aktiválás
270—360	ESC*: bitnyomtató üzemmód
310	nyomtatásimód-beállítás $0 <= \text{mód} <= 7$
330, 350	a nyomtatandó bitek száma (egy ciklusban 256)
370—400	első végrehajtáskor nem ugrik (így egyszerűbb a bitnyomtatást soronként beállítani)
410	a képernyő első betűhelyének „báziscíme"
420	a címet elteszem a verembe
440—450	a harmad képernyő 8 sor × 32 betűhely = 256 ciklus kell > IY+3, 32 betűhely > IY+2
490—520	készen van a 32 betűs sor?
530—550	a HL növelése és elmentése
560—570	készen van a 256 ciklus?
580	ha igen, jön a következő harmad
590	ha nem, újra
600—640	képernyő utolsó cím? Ha igen, VÉGE
650—700	ha nem, megnöveljük a címet 8 × 256-tal
720—730	a betűn belüli bitpozíció kijelölése, illetve növelése
750	a puffer növelése
760—780	a betű bitsorának utolsó bitje? Ha nem, újra
790—840	a pufferrekeszek tartalmának kinyomtatása
850—970	a regiszterek tartalmának elmentése, billentyűfigyelő rutin hívása, SYS+billentyű megnyomásra kilépés BASIC-be
990	IY számláló beállítása a 8 függőleges bit kiolvasásához
1010—1020	a HL betöltése a veremből
1030	az IY+1-be már előzőleg betöltöttük azt, hogy a regiszter melyik bitje van soron. Legyen ez a C bit
1050—1060	a C bit 0. pozícióba állítása
1070	a 0. kivételével a bitek törlése
1080—1120	a bit visszaléptetése
1130—1140	hozzáadjuk a pufferrekeszhez
1150—1170	kész a nyolc függőleges bit?
1190—1200	ha nem, vonjunk ki HL-ből 256-ot, mert alulról megyünk felfelé
1240—1320	a nyolc memóriarekesz a nyolc bitpozíciónak
1320—1350	a négy számlálórekesz

A programnak van egy még kiküszöböletlen, de már felfedezett hibája: csak egy képernyőtartalom nyomtatható ki, ezután a nyomtatót ki kell kapcsolni, mert különben a következő ábra eleje mindenféle „szemét" lesz. A nyomtató ki-, majd bekapcsolása után a következő ábra baj nélkül kinyomtatható.

Remélem, ha valaki más típusú nyomtatója miatt a programot egy az egyben nem is tudja használni, azért ötletet kap saját programjának megírásához.

DR. GRESZ MIKLÓS

```
10 ;HARDCOPY SPECTRUMRA
20 ;BROTHER M-1109 nyomtatora
30 ;Gresz Miklos 1987
40 ;
50        ORG  23300
60        LD   A,3
70        CALL #1601
80        LD   A,27
90        RST  16
100       LD   A,68
110       RST  16
120       LD   A,1
130       RST  16
140       LD   A,0
150       RST  16
160       LD   A,27
170       RST  16
180       LD   A,49
190       RST  16
200       LD   H,0
210 ESCBIT LD  A,13
220       RST  16
230       LD   A,10
240       RST  16
250       LD   A,9
260       RST  16
270       LD   A,27
280       RST  16
290       LD   A,42
300       RST  16
310       LD   A,0
320       RST  16
330       LD   A,0
340       RST  16
350       LD   A,1
360       RST  16
370       LD   (IY+2),32
380       LD   A,0
390       CP   H
400       JR   NZ,FOLYT
410 START LD   HL,18176
420       PUSH HL
430       LD   IY,CIM
440 SOROK LD   (IY+3),0
450       LD   (IY+2),32
460 HARM  LD   IX,PUFF
470       JR   BET
480 HAR   DEC  (IY+3)
490       DEC  (IY+2)
500       LD   A,(IY+2)
510       CP   0
520       JR   Z,ESCBIT
530 FOLYT POP  HL
540       INC  HL
550       PUSH HL
560       LD   A,(IY+3)
570       CP   0
580       JR   Z,VALT
590       JR   HARM
600 VALT  POP  HL
610       PUSH HL
620       LD   A,#58
630       CP   H
640       JR   Z,VEGE
650       POP  HL
660       LD   DE,256
670       LD   B,7
680 PLUSZ ADD  HL,DE
690       DJNZ PLUSZ
700       PUSH HL
710       JR   SOROK
720 BET   LD   (IY+1),0
730 BETU  INC  (IY+1)
740       JR   BIT
750 VISSZ INC  IX
760       LD   A,7
770       CP   (IY+1)
780       JR   NC,BETU
790 PRINT LD   IX,PUFF
800       LD   B,8
810 CIKL  LD   A,(IX)
820       RST  16
830       INC  IX
840       DJNZ CIKL
850       PUSH HL
860       PUSH BC
870       PUSH DE
880       CALL 654
890       LD   A,D
900       CP   24
910       POP  DE
920       POP  BC
930       POP  HL
940       JR   NZ,TOV
950       POP  HL
960       RET
970 TOV
980       JR   HAR
990 BIT   LD   (IY),9
1000      LD   (IX),0
1010      POP  HL
1020      PUSH HL
1030 OSZL LD   B,(IY+1)
1040      LD   A,(HL)
1050 LEP  RLCA
1060      DJNZ LEP
1070      AND  01
1080      LD   B,(IY)
1090 LEPES DJNZ TOVAB
1100      JR   TOVA
1110 TOVAB RRCA
1120      JR   LEPES
1130 TOVA ADD  A,(IX)
1140      LD   (IX),A
1150      DEC  (IY)
1160      LD   A,(IY)
1170      CP   2
1180      JR   C,VISSZ
1190      LD   DE,256
1200      SBC  HL,DE
1210      JR   OSZL
1220 VEGE POP  HL
1230      RET
1240 PUFF NOP
1250      NOP
1260      NOP
1270      NOP
1280      NOP
1290      NOP
1300      NOP
1310      NOP
1320 CIM  NOP
1330      NOP
1340      NOP
1350      NOP
1360      END
```



Informatizálás Bulgáriában és a Szovjetunióban

KLUBSZERVEZÉS ÉS OKTATÁS

Cikkem csak az átalakítás néhány jelével foglalkozik, melyeket a két országban tapasztaltam.

A szocialista országokban jó néhány év késéssel vették tudomásul, hogy a számítógépek használatának tömeges elterjesztése a mikroszámítógépek tömeges elterjesztésével jár együtt. Mi ennek felismerésében élen jártunk, és ez elsősorban a magánimportnak nevezett, alulról jövő kezdeményezésnek volt köszönhető. Ebben az úttörő szerepet, ahogy a világ számos más országában is, a klubok — nálunk az első és sokáig egyetlen HCC — töltötték be.

A címben említett két országban is az első lépést a klubok tették meg, miután tanulmányozták a mienket, de ott más úton jártak. A klubokat felülről szervezték, vagyis az állam hozta létre azokat, adta a felszerelést és a működtetés is rá hárul.

Bulgária az előtte álló úton már jóval előbb tart, mint a Szovjetunió. Burgaszban létesült talán a világon az első olyan klubépület, amelyet eleve erre a célra építettek, és amelynek egyetlen feladata is ez. A háromszintes épületben összesen 31 különböző típusú számítógépet — bolgár gyártású Apple II és XT kompatibilis gépeket — helyeztek el. Ezek nagy többsége díjmentesen használható bárki számára. Minden gépnek van háttértárolója (hajlékonylemez-egysége, néhánynak kis kapacitású merevlemezegysége), nyomtatójuk viszont jelenleg még nincs.

A klubok az üzemeltetési és továbbfejlesztési költségeket (a személyzet bérét, az energiaköltségeket kivéve) szoftvergyártással teremtik elő. Így például a Szófiai Városi Komputerklub animációs filmet készít a helybeli televízió részére. Vállalati felkérésre is készítenek szoftvereket. Ilyen megrendelésre ugyanez a klub készített szállodai nyilvántartó rendszert a szófiai szállodáknak.

A gépek elterjesztése érdekében Szófiában PC-szalont nyitottak. A mieinkhez hasonlóan a hazai gyártmányokat mutatják be és árusítják. Ami érdekes, magánszemélyek is vásárolhatnak a szalonban. (A forgalom valószínűleg nem lehet nagy, mert háromszor is jártam arra, de a boltban az eladókon kívül mást nem láttam.)

A gépválaszték az Apple II kompatibilis Pravetz 8 és az XT kompatibilis Pravetz 16. Érdekességként, azért, hogy a kettőt összehasonlítsam, megtudtam három konfiguráció árát. A Pravetz 16 256 k tárral, két hajlékonylemez-egységgel, egyszínű monitorral és az ezeket kiszolgáló illesztőkártyákkal 7800 leva (1 leva = 16 forint), ugyanez 10 Mbájtos merevlemezegységgel 12 500 leva, 20 Mbájtossal pedig 14 500 leva. Kapható még az Epson cég RX80 típusával kompatibilis nyomtató és számos szoftvertermék. Ez utóbbiak teljesen kezdőknek BASIC-anyagokból és professzionális felhasználóknak készítettekből (például PROLOG-ból) állnak. Áraikat nem tudom.

A Szovjetunióban nem országos szervek központi szervezésében jöttek létre az első klubok, hanem kutatóintézetek kisérleti vállalkozásaiként. Az első két, általam megismert klub a szibériai Novoszibirszkben, illetve a hozzá tartozó Akagyemgorodokban jött létre. Mindkét klub maga fejleszt és épít gépeket. Az elkészített gépeket, amelyekhez alkatrészt, eszközöket, helyiségeket a klub ad, jelképes összegért a tagok használatába adják. A gépeket építő szakember klubtagok az építést részben munkaköri kötelességként végzik. A klubokat a később alakítandó nagyszámú klub modelljeként kezelik.

Velünk ellentétben, náluk a klubok igyekeznek együttműködni, ezért létrehoztak egy szervezetet az első mintegy ötven klub részvételével. (Nem tudom megérteni, hogy nálunk sokszori felhívásom, az együttműködés előnyeinek ismertetése ellenére miért nem sikerült elérni az együttműködést a HCC és más klubok között. Ez annál is érthetetlenebb, mert a HCC és több külföldi — most már szovjet is — klub között élő kapcsolat áll fenn, ami elsősorban nekünk előnyös, és ezt a velünk együttműködni szándékozó hazai klubok is hasznosíthatnák.) Törekednek a nemzetközi együttműködés kialakítására is. Csatlakoztak felhívásunkhoz, ami a klubok nemzetközi szervezetének felállítására irányul.

A rendelkezésükre álló alkatrészválaszték alapján az egyik gépük 8080 típusú mikroprocesszorra épülő CP/M gép. A 64 k RAM tárú, árányalag jó grafikájú (320 × 256 pont, fekete-fehér) gépet hajlékonylemez-egységgel, monitorral együtt 1000 rubelért (1 rubel = 18 forint) kívánják forgalomba hozni. A másik két géptípus Spectrum 48 k kompatibilis. Mindkét gép közös tulajdonsága, hogy az ULA helyett TTL IC-ket alkalmaztak. Így az egyetlen import alkatrész a mikroprocesszor. A két változat között a legfontosabb eltérés a billentyűzet, ami az egyiknél a megszokott gumimembrán, a másiknál pedig normál elektromechanikus billentyűt használnak. Mindkét változat érdekes tulajdonsága a képernyőmemória lapozhatósága.

A két klubbal létrejött megállapodásban vállaltuk a kölcsönös segítségnyújtást, ami a mi részünkről — emlékezve az amerikai kluboktól kapott hasonló segítségre — elsősorban alkatrészek küldéséből áll.

Megtudtuk, hogy más gépekre, például C64-re alapozott klubok is léteznek. Ezekkel is igyekszünk kapcsolatba lépni.

A bolgár iskolahelyzetről már több cikkben is beszámoltunk. Változás elsősorban az XT kompatibilis gépek terjedése. A Szovjetunióban a Novoszibirszk-Akagyemgorodokban székelő Jerzsov akadémikus által vezetett kutatóintézetet bízták meg mind a módszerek kidolgozásával, mind azok gyakorlati bevezetésével. Ez igen nagy feladat, hiszen 140 000(!) iskoláról van szó.

A gépek szerint két szintet hoznak létre: az alsót a tömegesen alkalmazott, a magasabbat a terminálként is kapcsolható gépek alkotják. Tudomásom szerint az alsóra két típus a jelölt. Az egyik a japán Yamaha cég MSX2 szabványú gépe, lemezegységgel, monokrom monitorral, a másik pedig a francia Thomson cég MO5E gépe, 6809 típusú mikroprocesszorral. Az előzőhöz (és a Tandy CoCo, illetve a Dragon géphez) hasonló BASIC-kel, soros és párhuzamos csatornával, két analóg bemenettel. A magasabb szintet vagy a saját gyártmányú Apple II kompatibilis Agat típussal, vagy valamilyen XT kompatibilis géppel fogják betölteni.

Az oktatási célról, az oktatandó anyagról közelebbi információim nincs.

SIMONYI

ADOM A MAGYARÁZATOT!

Lapunk 1988/8. számában kérdeztük, hogy miért ad hibás eredményt az ott ismertetett egyenletrendszer-megoldó program.

A program írója figyelmen kívül hagyta azt a tényt, hogy a matematika korlátlan számtartománnyal dolgozik, a számítógépek azonban nem. Ebből két hiba adódik.

1. Minden BASIC-változatnál létezik egy, a változattól függő legkisebb pozitív szám (M1), amelynél kisebb abszolút értékűt már zérusnak tart a fordító. Ez az oka annak, hogy $0 < ABS(D) < M1$ esetén az egyenletrendszerre a program — tévesen — nem ad megoldást.

Bővítsük a Cramer-szabály egyenleteit 1/M1 értékkel. Például: $y = \frac{DY}{D} = \frac{DY/M1}{D/M1}$

Ha egyenletünk nevezőjére igaz, hogy $M1 < ABS(D/M1) < 1$, akkor adódnak téves eredmények. Nem kapunk azonban ilyenkor sem téves eredményt, ha a bővített egyenletekkel dolgozunk.

2. Létezik egy legnagyobb szám (M2). Ennél a számláló bővítés nélkül is túl nagy lehet. Például $ABS(DY) > M2$. Ilyenkor, ha a nevező egynél kisebb abszolút értékű, a keresett változó — ami nálunk y — is nagyobb lesz M2-nél. Ebben az esetben az eredeti program a BASIC „túlcsordulás" hibaüzenetével leáll!

Bekövetkezhet ez akkor is, ha megengedhető nagyságú számokat adunk be? Igen, például másodrendű egyenletrendszernél $DY > M2$, ha történetesen $M2 > a(1) > 1 + SQR(M2)$

$M2 > b(2) > 1 + SQR(M2)$

$0 < a(3) < 1$

$0 < b(1) < 1$ mert ekkor

$DY = a(1)b(2) - [a(3)b(1)] > [1 + SQR(M2)]^2 - -1^2 = [2 + SQR(M2)]\ SQR(M2) > M2$

Az elözökben ismertetett két probléma együtt is jelentkezhet. Nézzük meg a lehetséges eseteket:

1. *Egyik sincs.* Erre az esetre jó az eredeti program.

2. *Csak az első van.* Erre jó, ha az 1/M1 értékű bővítést alkalmazzuk, kivéve, ha a bővítéstől a számláló lesz túl nagy, tehát ha $ABS(DY/M1) > M2$. Ilyenkor mint a harmadik eset.

3. *Csak a második van.* Erre az esetre jó, ha 1/M2 értékkel bővítünk. Kivéve, ha a bővítéstől a nevező lesz túl kicsi. Ilyenkor mint a második eset. Ez alól is van azonban kivétel. Amennyiben először volt a második eset, abból lett a harmadik, és aztán újra a második lenne. Ekkor olyan, mintha a negyedik eset lenne.

4. *Mindkettő van.* Ez is bekövetkezhet, például másodrendű esetnél, ha y-t keressük és

$a(1) = 1$

$a(2) = 2.1 \cdot M1$

$a(3) = 1$

$a(4) = 3 \cdot M1$

$b(1) = -M2/1.8$

$b(2) = M2/2$ akkor

$DY = 19 \cdot M2/18 > M2$

$D = 0.9 \cdot M1 < M1$

Ha itt bővítünk, akkor

$DY = 19/(18 \cdot M1)$

$D = 0.9/M2$

A legtöbb Microsoft BASIC-nél

$M2 \sim 1/(2 \cdot M1)$ így

$D \sim 1.8 \cdot M1 > M1$ de

$DY \sim 19 \cdot M2/9 > M2$

Tehát a probléma általában bővítéssel oldható meg. Kérdés, hogy szükséges-e ez egyáltalán? Ennél az esetnél $DY > M2$, $D < M1$, így $Y > M2/M1$, ami tehát a BASIC-fordító számára „túlcsordulás". Így ennél az esetnél számítás nélkül azt kell kiiratni, hogy "NAGYOBB MINT";M2/M1

A cikk elején említett probléma megoldására szolgáló programnak a négy eset mindegyikére kiterjedő megoldást kell tartalmaznia. Ez azt jelenti, hogy az eredeti programot úgy kell módositani, hogy az eseteknél említett megoldásokat is tartalmazza. Egy ilyen megoldást ismertetek a következő számban.

S. E.

μINFORM

Rovatunkban az Apple, Atari, Commodore és Sinclair mikrók tulajdonosait feltehetően érdeklő, angol és német nyelvű cikkekről információkat olvasóinkat egy tartalomleíró szólánc segítségével.

A forráshely karaktersorozatát nyíl vezeti be, ezt a / jelig a folyóirat kódja követi (lásd táblázat). A két / jel között a megjelenési adatokat (év, hó), illetve a cikk kezdő oldalszámát szerepeltetjük. A második / jel után pedig - az esetleges másolatkérést megkönnyítendő - a cikk teljes oldalterjedelmét közöljük.

A folyóiratok megtekinthetők a SZÁMALK (Bp. XI. Szakasits Á. u. 68.), illetve - a x-gal jelzettek - az OMIKK (Bp. VIII. Múzeum u. 17.) szakkönyvtárában. (A másolás díja oldalanként 8 Ft.)

A Magazin helyhiány miatt mindössze egyetlen címszó, a programlista közzétételére vállalkozhatott a folyamatosan bővített adatbázisból. A kedvező visszhang alapján az OMIKK háromhavi bontásban kiadja a tartalomleíró szólánc permutálásával és alfabetikus rendezésével szerkesztett teljes anyagot. Az "APACS Mikroindex" első füzete már megjelent, ára 54,- Ft.

Több példány vásárlása/rendelése esetén 10 darabonként 2 tiszteletpéldányt térítésmentesen ajándékoz a terjesztőknek az OMIKK vevőszolgálata (Bp. Pf.: 12. 1428).

A folyóirat neve	Kódja
x 64'er Magazin	64er
Antics	anti
x Chip Magazin	chip
x Compute!	cute
x Dr. Dobb's Journal	dobb
Elektor Electronics	etor
Happy Computer	happ
x mc - Zeitschrift	mc
Run (USA)	run
Run (NSZK)	run2
x Your Computer	your
x ZX Computing Monthly	ZXCM

PROGRAMLISTA
adatbazis/c64/filekezeles/30000 karakteres szekvencialis allomanyok/masztechnika/<sdat> ->run2/87.09-102/5
PROGRAMLISTA
amiga/kepernyokezeles/menu-parancs bovito utasitasok/<supermenus>
->cute/87.09-88/4
PROGRAMLISTA
animacio/atari xl/xe/4.resz/demo
->anti/87.09-20/6
PROGRAMLISTA
atari st/programiras/basic/forrasfile beillesztes gem-rutinokkal
->happ/87.09-105/2
PROGRAMLISTA
atari xl/xe/f-billentyuk programozasa/<keyboard 2000> ->happ/87.09-66/2
PROGRAMLISTA
atari xl/xe/jatek/<attack on the doomstar> ->anti/87.09-15/4
PROGRAMLISTA
atari xl/xe/jatek/<be the eggman>
->anti/87.09-35/5
PROGRAMLISTA
atari xl/xe/jatek/oktatas/kviz-file keszites/valaszelemzes
->anti/87.09-53/4
PROGRAMLISTA
atari xl/xe/kamatszamitas/<annual % rate> ->anti/87.09-40/7
PROGRAMLISTA
atari xl/xe/levelezes/cimlista osszeallitas/nyomtatas ->anti/87.09-30/6
PROGRAMLISTA
atari xl/xe/oktatas/matematika/'szabalyos'-modusu fix/lebegopontos aritmetika ->anti/87.09-26/6
PROGRAMLISTA
atari xl/xe/programiras/basic/karaktersorozat-kereso rutin
->cute/87.09-84/2
PROGRAMLISTA
c128/boot-funkciok kiegeszitese/<boot-sektor-manipulator>
->run2/87.09-49/3
PROGRAMLISTA
c128/filekonverzio/grafika/c64-forma fun illesztese c128-modushoz
->run2/87.09-39/3
PROGRAMLISTA
c128/lemezkezeles/<128 notepad>
->run/87.09-48/4
PROGRAMLISTA
c128/programiras/f-billentyuk atprogramozasa ->run2/87.09-43/2
PROGRAMLISTA
c64/bioritmus-grafikon keszites
->64er/87.09-109/5
PROGRAMLISTA
c64/grafika/mandelbrot/1.resz/alapfogalmak/peldak ->64er/87.09-61/7
PROGRAMLISTA
c64/grafika/programiras/color/screen-ram cimek megjelenitese kurzorral/<screen genie> ->run/87.09-68/2
PROGRAMLISTA
c64/jatek/<duell> ->64er/87.09-38/3
PROGRAMLISTA
c64/jatek/<falling ball>
->run2/87.09-95/6
PROGRAMLISTA
c64/jatek/<mini breakout>
->run2/87.09-118/3
PROGRAMLISTA
c64/jatek/<phaser phire>
->run/87.09-58/4
PROGRAMLISTA
c64/jatek/<topsy-turvy>
->happ/87.09-51/7
PROGRAMLISTA
c64/lemezkezeles/f-billentyuk programozasa ->run/87.09-63/2
PROGRAMLISTA
c64/mailbox/sysop parameterek hasznalatellenorzes/pelda
->happ/87.09-139/1
PROGRAMLISTA
c64/matematika/grafikak keszitese/<hiepfer> ->64er/87.09-57/4
PROGRAMLISTA
c64/mesterseges intelligencia/dedukciv gondolkodas szimulacioja/<smart alec> ->cute/87.09-77/5
PROGRAMLISTA
c64/nyomtatas/33.9mmx23.5mm-es hires grafikak keszitese
->happ/87.09-57/2
PROGRAMLISTA
c64/programiras/assemblerrel segitett basic6.resz/pull-down menuk
->64er/87.09-68/8
PROGRAMLISTA
c64/programiras/gepi kod/segedfunkciok beepitese az operacios rendszerbe/<c64+> ->cute/87.09-97/3
PROGRAMLISTA
c64/programiras/linker a cimgorgetoszinvarialo eloprogramhoz/lasd (87.07.) ->happ/87.09-60/2
PROGRAMLISTA
c64/programiras/uzenetszovegi/<letter-maker> ->run2/87.09-101/1
PROGRAMLISTA
jatek/plus/4/<boulder-dash> adaptacio ->run2/87.09-126/4
PROGRAMLISTA
ram/atari xl/xe/bovites/programiras/<hidden ram> ->cute/87.09-91/2
PROGRAMLISTA
c128/lemezkezeles/<128 notepad>
->run/87.09-48/4
PROGRAMLISTA
c128/programiras/f-billentyuk atprogramozasa ->run2/87.09-43/2
PROGRAMLISTA
c64/bioritmus-grafikon keszites
->64er/87.09-109/5
PROGRAMLISTA
c64/grafika/mandelbrot/1.resz/alapfogalmak/peldak ->64er/87.09-61/7
PROGRAMLISTA
c64/grafika/programiras/color/screen-ram cimek megjelenitese kurzorral/<screen genie> ->run/87.09-68/2
PROGRAMLISTA
c64/jatek/<duell> ->64er/87.09-38/3
PROGRAMLISTA
c64/jatek/<falling ball>
->run2/87.09-95/6
PROGRAMLISTA
c64/jatek/<mini breakout>
->run2/87.09-118/3
PROGRAMLISTA
c64/jatek/<phaser phire>
->run/87.09-58/4
PROGRAMLISTA
c64/jatek/<topsy-turvy>
->happ/87.09-51/7
PROGRAMLISTA
c64/lemezkezeles/f-billentyuk programozasa ->run/87.09-63/2
PROGRAMLISTA
c64/mailbox/sysop parameterek hasznalatellenorzes/pelda
->happ/87.09-139/1
PROGRAMLISTA
c64/matematika/grafikak keszitese/<hiepfer> ->64er/87.09-57/4
PROGRAMLISTA
c64/mesterseges intelligencia/dedukciv gondolkodas szimulacioja/<smart alec> ->cute/87.09-77/5
PROGRAMLISTA
c64/nyomtatas/33.9mmx23.5mm-es hires grafikak keszitese
->happ/87.09-57/2
PROGRAMLISTA
c64/programiras/assemblerrel segitett basic6.resz/pull-down menuk
->64er/87.09-68/8
PROGRAMLISTA
c64/programiras/gepi kod/segedfunkciok beepitese az operacios rendszerbe/<c64+> ->cute/87.09-97/3
PROGRAMLISTA
c64/programiras/linker a cimgorgetoszinvarialo eloprogramhoz/lasd (87.07.) ->happ/87.09-60/2
PROGRAMLISTA
c64/programiras/uzenetszovegi/<letter-maker> ->run2/87.09-101/1
PROGRAMLISTA
jatek/plus/4/<boulder-dash> adaptacio ->run2/87.09-126/4
PROGRAMLISTA
ram/atari xl/xe/bovites/programiras/<hidden ram> ->cute/87.09-91/2
PROGRAMLISTA
ram-floppy/c128/segedprogramok az '1700/1750 expansionmodul'-okhoz
->run2/87.09-51/5
PROGRAMLISTA
ram-floppy/c128/vdc-8563 atprogramozasa/<vdc-ram-floppy>
->run2/87.09-46/3
PROGRAMLISTA
spectrum/programiras/basic compiler/<uschi>/alkalmazasi utmutato
->happ/87.09-68/4
PROGRAMLISTA
szovegfeldolgozas/plus/4/script/plus-modul modositas/kiegeszites specialis karakterekkel ->run2/87.09-123/3
PROGRAMLISTA
zene/apple ii/kompaktlemez/rogzites a kazettaporton keresztul/lemezreirt adatok 'playback'-je
->cute/87.09-86/2
PROGRAMLISTA
zene/c64/grafika/programiras/48 uj basic utasitas/'poke'-helyettesites
->run2/87.09-109/9
PROGRAMLISTA
zene/c64/hangkimenet/digitalizalas audiokazettarol a 'datasette'-tel
->run2/87.09-107/2

Egyesek úgy vélik, hogy a számítógép-tudomány (a számítástudomány) alig több, mint programozás — nem érdemes önálló diszciplinaként tanítani. A szóban forgó cikk írója 1981-ben nem hivatalosan megkérdezte a számítástudomány több előadóját: saját tizenéves gyermekének fő tantárgyaként ajánlaná-e saját szakterületét. Egyikük sem ajánlotta. Inkább a matematikát, a gazdaságtant, a fizikát stb. javasolták volna. Csak utána jöhetett volna a számítástudomány. A megkérdezettek egyébként a számítástechnikát mint pályát, mint hivatást, foglalkozást természetesen mind jónak tartották, de úgy gondolták, amit ma tanítanak ezen a címen, az hamar, gyorsan elavul. Gyakran hivatkoztak arra, hogy a számítástechnikai (-tudományi) végzettségűek a végzés után néhány évvel nem produkálnak ezen a területen többet, mint a más szakon végzettek.

Az idézett cikk szerzője azt állítja, hogy kialakítható olyan filozófia, összeállítható olyan tanterv, mely alapján önálló diszciplinaként tanítható a számítástudomány. Ez a „tudás—információ—adat" hármas fogalomcsoportja köré szervezhető. De mi az adat? Mi az információ? Azon kívül, hogy azt mondjuk: ezek a valóság valamilyen reprezentánsai, letérképezései, a válasz a problémát elemző ember szemléletétől, illetőleg attól az absztrakciós szinttől függ, amelyen az illető dolgozik. Például egy biológiai rendszer meghatározó tulajdonságai nem feltétlenül jelennek meg a kémia alapvető szintjén, bár elvileg megmagyarázhatók lennének a kémia és a fizika fogalmaival. Azt lehetne mondani, hogy történetesen a szociológia, a pszichológia, a biológia, a biokémia, a kémia és a fizika valamilyen hierarchiát alkot. Nem kell mindenkinek feltétlenül egyetértenie egy ilyen hierarchiával, de ez most nem is fontos. Az idézett cikk szerzője szerint az információs rendszerek tervezésénél valamilyen általánosan elfogadott megközelítésből kiindulva el lehet jutni valamilyen megfelelő hierarchiához. Példákat is hoz fel erre:

információs szint; koncepcionális szint; fizikai szint vagy a teljes rendszer szintje; információs szint, az adatok szintje; a fizikai reprezentáció szintje, avagy tudás, információ, nyelv; adatok; fájlok, rekordok; jelek; számítógépes ábrázolás, karakterek, szavak, fájlok, elektronika (és manapság száloptika?).

Bár rovatunkban magyar nyelvű publikációkkal nem szoktunk foglalkozni, most mégis kivételt teszünk, mert a témához szorosan kapcsolódó alapvető publikáció jelent meg a Magyar Tudomány 88/7—8. számában: „A számítástechnika és a számítógép-tudomány értékrendi kérdései" címmel. Mivel ez az írás Magyarországon viszonylag könnyen hozzáférhető, ezért ízelítőül sem közlünk belőle részleteket, csak minden téma iránt érdeklődő olvasónk szíves figyelmébe ajánljuk.

Augusztus óta rovatunk minden egyes cikkében teljes egészében közöltünk egy-egy Updike-írást is. Következzék most is egy — fogadják szeretettel.

J. Updike: Vízvezetékeink

Az öreg szerelő gyöngéden előrehajol, nemrég vett házam pincéjének félhomályában megmutat egy mives, antik csőkötést.

— Már vagy harminc éve nem csinálják így — mondja. Hangja vékonyan szivárog, akár a rozsdán átpréselődő víz. — Harminc-negyven éve. Mikor elkezdtem dolgozni apám mellett, még csak így csináltuk. Régi ólomkötés. Kenni kellett: egyik kezemben forró öntőkanál, másikban vizes rongy. Tizenhat mozdulatot kellett megtenni, mielőtt az ólom kihűlt. Tizenhat különböző mozdulatot. Másképp vége, a kötés tönkrement, le kellett az egészet vésni, kezdhette az ember elölről. Amikor nekivágtam a szakmának, így kellett csinálni. Lehettem vagy tizenöt-tizenhat éves. Ez a kötés, itt ni, lehet vagy ötvenéves.

Ő ismeri csöveimet, én csupán birtoklom őket. Tulajdonosok jönnek-mennek, ő marad. Azt hisszük, tudjuk, látjuk, kik vagyunk, aztán rájövünk, valójában csupán egy rakás szerencsétlenség vagyunk. Úgy véljük, életet, szép kilátást vásároltunk, mikor valójában csupán labirintust, történelmet, csőlelleteket, csatlakozásokat, búzelzárókat, szelepeket. A szerelő egy vastag, sötét csövet mutat, mely átlósan az alapba fut.

— Látja ezt a vonalat a cső alján? — Fehér vonal, leheletnyi lerakódás, fakó oxid.

— Ne nyúljon hozzá, csepegni fog. Látja, ezt a csövet két részből öntötték, úgy kellene beépíteni, hogy a varratok oldalra nézzenek. De néha varrattal lefelé építik be őket. — Két begömbölyített tenyerével mutatja, hogyan. A két tenyér lassan szétnyílik — tágul a rés. Markai között kilesve magam is szinte kiutat kereső vízzé válok ügyes bemutatásában.

— Látja, a végén elkezd csöpögni. — Zseblámpájával visszafelé követi az árulkodó fehér vonalat.

— Négy-öt új szakasz megtenné. — Felsóhajt, zihál. Pupillája a félhomályban töltött élet miatt tágabb, mint az átlagemberé. Költő: ahol én csupán hibát, bosszantó kiadást előidéző tökéletlenséget látok, ott ő gyöngéden pillogat, elmereng a korrózió és az áramlások örökkévalóságában. Ironikus számlaremekműveket küld nekem, benne helyes kis alkatrészek jegyzékével:

1 db 1 1/4 × 1"	galv. karmantyú 58 cent
1 db 3/8"	sárgaréz visszacsapó szelep 90 cent
3 db 1/2"	csőkötés 13 cent.

Szinte őrültnek tűnik e kínos precizitás, melyet a végén nyel le egy mindent elsöprő „munkadíjnak" nevezett szép, kerek összeg:

munkadíj 550 dollár.

Feltételezem, mostani elérzékenyült meditációit — a hosszú, pillogásokkal kitöltött szünetekkel együtt — szintén munkának számítja.

Régi házunk, melyből elköltöztünk, egy mérföldnyire innen, szinte megkönnyebbült, hogy megszabadult tőlünk. A szobák — valamikori étkezéseink, ceremóniáink, magándrámáink szinterei, hol egyesek közülünk felserdültek, ahol a tér, a lépcsők telítődve mindennapi mozdulatainkkal és szabálytalanságaikkal úgy csontjainkba ivódtak, hogy éjszaka is baj nélkül tudtunk közlekedni bennük — várakozásom ellenére nem látszanak gyászolni minket. A ház ujjong hirtelen visszanyert méretein, üres sarkai tágasságán. A sokáig szőnyeggel takart padlódeszkák ragyognak, mintha frissen lakkozták volna őket. A nap háboritatlanul süt be a függönyüket vesztett ablakokon. Megfiatalodott a ház. Amíg az új tulajdonosok meg nem érkeznek, ismét saját maga lehet, ismét saját életét élheti, melyet egy időre eltakart a miénk. A padlódeszkákat most csak a holdvilág recsegteti. Ha néha reggelente visszatérek egy-két ottfelejtett apróságért — kandallóvasért, képkeretért —, szűzies szemtelenséggel fogad. Ha ajtaját kitárom, mintha csak a macskát engedném be reggeli tejéért: miákol egyet éjszakai álmunk melegét még őrző ágyunk felé haladtában. Szokásai — e nyávogás és a megosztott közös fedél erejéig — hajszálfinoman illeszkedtek a mieinkhez. A természet nem annyira érzelgős, mint ahogy azt az ekológusok feltételezik. A ház mindössze egy nap alatt elfelejtett bennünket.

Bűntudatot érzek, hogy oly kevéssé vettük birtokunkba: három szállítómunkás meg egy fuvallat ki tudta törölni még az emlékünket is. Jó tíz éve, amikor beköltöztünk, nem volt kisértetlakta, bár gerendái és a kandallók lehettek vagy háromszáz évesek. Azt hittem, kora miatt az lesz, de egy amatőr boszorkány, akit feleségem még az egyetemről ismert, megkopogtatta a hálószobák falait, körülszaglászott a padlásszobában és biztosított bennünket, hogy a hely kisértetmentes. Szemei (nem furcsa?) ugyanolyan természetellenesen tágra nyíltak, mint a szerelőé. A házat puritán farmerek építették. A XIX. században vendégfogadóként szolgálhatott: a newburyporti bekötőút itt vezetett el nem messze. S a harmincas években bérházzá alakították. A ma oly tágas szobákat gipszkarton falakkal megosztották, a falakon lyukakat vágtak, ezen a lakók cukrot, lisztet cserélhetek. Falusias idők, szegény idők. Egy időben még csirkéket is tarthattak a padláson. Kezdetben a gyerekek azt mondogatták, esős időben érzik a nedves toll szagát, de ezt csupán a fantázia és a mitosz hatalmának tulajdonitottam. A kertet ásva, egy letűnt kor technikájának emlékeiként, néha önkanalak és furcsa törött flaskák kerülnek elő a földből. Utánunk mindössze néhány golflabda marad az iriszek között, és néhány poros teniszlabda a fütőtestek alatt. Hátrahagyott szellemeinket csak mi magunk látjuk.

Egy férfit látok szmokingban és egy asszonyt hosszú fehér ruhában, amint ide-oda lépegetnek a kertben. A hideg szél nevetésre ingerli őket. Hajnali kettő van, húsvét hajnala. Fóliába burkolt csokoládétojásokat rejtenek el mindenfelé, be vannak csipve. Reggel selymesen bizsergető fejfájással a csokoládéra vadászó gyerekek kiabálására és veszekedésére ébrednek, amint azok maszatos szájjal és émelyítően édes lehelettel bemásznak szüleik ágyába. De a hajnali jelenés az, amit a konyhában állva a kijózanodott tudat perspektivájából látok, a féktelen kedvű párt látom, amint ott ügyetlenkednek a kert sarkában, az aranyesőbokor és a hinta körül — két húsvéti nyuszi.

Férfi hajol egy gyerekágy fölé, együtt imádkozik a gyerekekkel. Bajban vannak az imádság szövegével — más-más vasárnapi iskolába járták.

egy cikket a számítástudomány és az informatika tanításának filozófiájáról (L. Capper: „A Philosophy for the Teaching of Computer Science and Information Technology" the Computer Journal, Vol. 29. No. I., 1986.) és erről eszünkbe jutottak azok a fiatalok, akik már megismerkedtek a számítógéppel, a BASIC nyelvvel, megszerették a számítástechnikát és azt fontolgatják, hogy válasszák-e a számítástechnikát egy életre hivatásul. A cikk központi kérdése: tudomány-e, tudományosan megalapozható-e a számítástechnika — vagy másképpen: kellően megalapozható-e a számítástechnika oktatása.

A férfi elgyötörten és (talán a csirketollak szelleme miatt?) kissé asztmásan átmegy a szomszéd szobába, miközben már az jár eszében, hogy le kellene menni elolvasni egy könyvet, meginni egy pohár italt. Ott a gyerek — egy nagyobb — halkan felkiált, amikor felajánlja neki, hajtsák együtt imára fejüket. „Apa, ne!" Kerek fehér arca — csak homályosan kivehető az esti félhomályban — szinte átizzik a feszültségtől, a zavartól, az esdeklástől. A férfi maga is zavarban van — túl könnyen zavarba lehet hozni —, megcsókolja a lányát, kihátrál, becsukja a hálószoba ajtaját, otthagyja őt a sötét szobában.

A legnagyobb szoba falai — eltekintve azoktól a kísértetszerű négyszögletes foltoktól, ahol valaha a könyvszekrények álltak, képek lógtak —, most csupaszok. Itt vitatkoznak, szenvedélyesen gesztikulálnak. Az asszony, a feleség, éppen elhajít valamit — eredeti szándéka szerint egy hamutartót, de a rendszeretet még ilyen rózsavörös arcú, feldúlt állapotban is győzedelmeskedik benne, és végül is egy könyv mellett dönt. Könnyekben tör ki, talán puritán gyökerű képtelensége miatt, hogy egyszer jól odavágjon egy hamutartót. Átrohan egy másik szobába, és nem felejti el átugrani a küszöböt sem, melyben idegenek gyakran megbotlanak. A gyerekek közben csendesen fel-le lopakodnak a lépcsőn. Sápadtak, tele vannak bűntudattal, ártatlan kis szívük rejtekén magukat hibáztatják e zűrzavarért. Még a kutya is összekuporodik, farkát maga alá húzza. A férfi a heverőbe süpped, mely már nincs ott a helyén. Bokái összezárva, feje leszegezve, mintha bilincsbe lenne verve. Túljátssza helyzetét, úgy tesz, mintha börtönbe lenne zárva. Valószínűleg nyár van, mert egy kis káposztalepke oda nem illően fehérlik a zsalugáteren, melyhez kívülről mályvarózsák dörzsölődnek-ütődnek gyengéden. Az asszony visszatér, arca vörösből már rózsaszínre váltott, most már szabatosan, megfontoltan érvel. A férfi feláll és kiabál. Az asszony megüti, az félrelöki a másik kezét és oldalba vágja. Meglepődik, milyen kellemesen szivacsos anyagot ér: egy zsák zsiger. Ott ugrándoznak az útjukba eső bútorok között, amikből apró porfelhők szállnak fel. A gyerekek egy lépcsőfokkal feljebb húzódnak. A kutya begörbíti a hátát, mintha vernék, az ajtóhoz megy, kikéredzkedik. A férfi átöleli az asszonyt és suttog neki valamit. Az rózsaszín, felhevült és folynak a könnyei. A férfi észreveszi, hogy ő maga is sír, milyen jó érzés, mint amikor végre hányni tud vagy jól kiizzad az ember. Mit mond egymásnak, miről beszél egymással ez a két erőszakos, ijedt ember? Változásról, természetes folyamatról, az idő múlásáról, a halálról.

Gyarló szellemek. Eltűnnek, mint a lehelet nyoma az ablaküvegről. Ellenpontként emékezetembe idéződnek gyermekkorom hatalmas, súlyos, szaftos kókuszreszelékkel töltött húsvéti tojásai. Nehezek, mint egy öntvény, vagy inkább tágasak, mint egy színház, telis-teli papírkivágással — bennük egész kis sziluettvilág, mely saját napfényt áraszt. Ezek a tojások lila forgácsfészkükben felhozzák a rejtelmek csillagokat ásztató, szivárgásmentesen soha le nem zárható kútjából azokat az időket, amikor azok a réges-régi, születésem előtti fényképek készültek, amikor Isten még odafigyelt ránk. Vasárnap reggel van, szinte még éjszaka. Imádkozom. Úgy fekszem egy sötét, sarkaiban szellemlakta házban, mint egy szalmaszál a szakadék peremén. Walt Disney-ihletésű, begörbített karmú szörnyek vesznek körül. Egy városban vagyok, melynek legfőbb büszkesége a központjában elterpeszkedő ravatalozó, melyet köröskörül boszorkányjelekkel megjelölt pajták fognak körül. A szalon szőnyegén egy kontinens alakú folt éktelenkedik, pici koromban odahánytam. Mitosz mitoszon: már három- vagy négyéves vagyok, éhes, mindenféle piszkot eszem furcsa, tollbokrétaszerű, ködös, trópusi páfrányokat tartó virágcserepekből. Egyik nagyanyám babonás hite szerint a gyereknek évente legalább egy font piszkot kell megennie, attól lesz erős. Később — kilenc-tíz éves koromban — a hasamon fekszem ugyanott, az újságot olvasom fel vak nagyapámnak. Először a halálozási rovatot, aztán a helyi híreket, végül az első oldalról a főcímeket a japánokról és Rooseveltről. A papírnak jellegzetes szaga van, nem áporodott, mint a képregényeké. Frissebb, kevésbé édeskés, mint a fánkos zacskóé, de fűszeresebb, izgatóbb, benne van a jövő, a ropogósan, frissen, a kissé még melegen csomagolt dolgok illata, az *újdonságoké*. Észreveszem, ez a szag naponta megérkezik és el is enyészik. Aztán tizenhárom éves leszek, és elbúcsúzunk a szalontól. Költözünk. A kontinens formájú folt mellett ott virítanak a páfrányok cserepeinek kerek lenyomatai is. A függöny nélküli albakokon zavartalanul besütő nap szinte felfedezi őket. Olyan mélyek, mint a dinoszauruszok lábnyomai.

Gyerekeink érzékelték-e vajon a mi hajnali húsvéti nyuszisköadásaink frivolságát? A fiatalabbik ujját szopva a kisebb szobában feküdt, és mellettem a sötétbe bámult valamire. Számára házunk úgy át lehetett itatva rettenettel, hogy az emlékezetének minden rétegére rányomhatta bélyegét. Neki minden apró karcolás a falon megfejthetetlen jelentéssel telitődhetett. Ő volt az egyetlen gyerek, aki beszélni szokott a halálról. Holnap volt a születésnapja.

— Nem akarok születésnapot. Nem akarok kilencéves lenni.

— De nőnöd kell. Mindenki felnő. A fák is.

— Nem akarok.

— Nem szeretnél olyan nagy lány lenni, mint Judit?

— Nem.

— Kifesthetnéd magad, melltartót viselhetnél, biciklizhetnél a főutcán.

— Nem akarok biciklizni a főutcán.

— De miért nem?

— Mert a végén megöregszem és meg fogok halni.

És patakzanak a könnyei, és a férfinak mellette erre már nincs szava, ahogy egyetlen férfinak sem lenne ebben a kis szobában, ahol semmi sem marad utánunk, csupán néhány karcolás és az ablakkereten egy félig levakart, kifakult Maci-embléma. Ha nem költöztünk volna el, ideje lett volna már feltenni a téli védőtáblákat az ablakokra.

Nyílik az őszi kikerics régi házunknál, az újnál virágoznak a nárciszok. A gyerekek, akik előttünk laktak itt, teniszlabdákat rejtettek el számunkra a radiátorok alatt — meglepetésül. Amikor alkudtunk az új házra, észre-észrevettük őket, amint ott bujkáltak házuk körül, és a bokrok, a kerítés mögé rejtőzve bámultak minket, jövendőjük bitorlóit. Miután ők már kiköltöztek, és mi még nem költöztünk be, bolondos, féktelen játékokat játszottunk az üres szobákban, a falaknak pattintottuk a labdákat, össze-vissza ugrabugráltunk. A labdák hamarosan újra elvesztek. A szobák megteltek bútorral.

A szerelő gyengéden elmereng, és egy lefűrészelt csőcsonkot mutat, a kútból a nyomótartályba vezető szakasz egy darabját. A cső belső átmérője az ásványi lerakódásoktól nem nagyobb, mint az ujja. A koncentrikus rétegek olyan vékonyak, akár a papír. Az egész egy végétől kezdve olvasandó könyv képzetét kelti, olyanét, melyet nem szokás kinyitni, amit papok őriznek bölcsen, lakat alatt.

— Figyelje csak — mondja —, mindez negyven-ötven év alatt rakódott le. Emlékszem, mikor apámmal beszereltünk ide egy szivattyút, ez a cső már itt volt. Nem lehet ez ellen semmit se tenni, az oldott ásványi anyagok miatt van. Nincs mit tenni, ki kell ásni és ki kell cserélni egy új ötnegyedesre vagy hatnegyedesre.

Magam elé képzelem, amint a pázsitomat felszaggatják, amint a nagy, aranyszínű markológép letapossa a nárciszaimat, amint dollárjaim szépen elfolydogálnak. Tiltakozásom hatástalan.

A szerelő felsóhajt, ahogyan a költők szoktak, fél szemmel a hallgatóságra sandítva.

— Nézze, ha így hagyja, egy szép napon leég az új szivattyúja. Túl nagy munkát kell kifejtenie, hogy felszivja a vizet. Cserélje ki most, soha többé nem lesz gondja vele. Túléli magát.

Engem, meg őt. Szemei tágra nyílnak a korrózió és az áramlás néma jelenlétében. Keresztülpréseljük magunkat a pincefeljárón; az égbolt egy vakító darabja foglalja el helyét felettünk ideiglenes-időtlen felhőkkel tarkítva. Körös-körül — minden túlél minket.

(Fordította: K. E.)

From the book Museums and other stories by John Updike.
© 1971, 1972, 1975, 1983 by John Updike.
„Plumbing" originally appeared in The New Yorker.

ANYAG ÉS MOZGÉKONYSÁG

A sakkhadállások értékelésének két legfontosabb tényezője az anyag és a mozgékonyság. Ezeknek az értékelési módjaival régebbi számainkban már megismerkedhetett az olvasó. Ezúttal a velük kapcsolatos statisztikai adatokat ismertetem, amelyeket Dap Hartman dolgozott fel a legutóbbi alkalommal már említett, 832 nagymesterjátszma számítógépes elemzésével. A játszmák feldolgozásánál az anyagi érték számítását Shannon 1950-es definíciója alapján végezte el. Ezek szerint:

$F_{anyag} = 0{,}9*V + 0{,}5*B + 0{,}3*F + 0{,}3*H + 0{,}1*GY$,

ahol V a vezérek, B a bástyák, F a Futók, H a huszárok, GY a gyalogok száma. Ezek szerint az alapállásban világosnak és sötétnek egyaránt 3,9 anyagi összértékű figurája van.

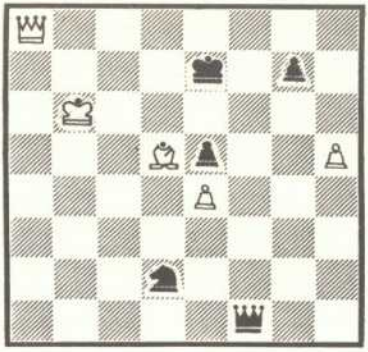
Az *1. ábrán* azt láthatjuk, hogyan alakul az anyagi érték a féllépésszám függvényében. Szembetűnő, hogy a tizedik féllépésig szinte stagnál, majd rohamosan csökken a 129-edik féllépésig, amelytől viszont kismértékben, de emelkedik.

Az anyagi érték változásának ez a három szakasza jól szemlélteti a sakkjátszma három fázisát. Az első szakaszban, a megnyitásban, mindkét fél kifejlődik tisztjeivel, ezért kevés a cserelehetőség. A második szakaszban, a középjátékban viszont a kombinatív játék kerül előtérbe, ami a figurák lecserélését is magával hozza. A leegyszerűsítések után a középjáték fokozatosan átmegy a végjátékba. A 129-edik féllépés után pedig a gyalogátalakulások miatt ismét nő világos és sötét anyagi értéke.

A *2. ábra* világos és sötét anyagi értékének a különbségét mutatja a féllépésszám függvényében. Látható, hogy a két fél anyagi értékének a különbsége általában akkor a legnagyobb, amikor anyagi értékük összege a legkisebb. Ez ellentmondásnak tűnik, de ha szemügyre vesszük a 2. ábrát, akkor láthatjuk, hogy a két fél anyagi értéke közötti legnagyobb különbség sem nagyobb kétgyalognyinál.

Ez azért van, mert a nagymesterek kiegyensúlyozottan játszanak, és a legkisebb pozíciós előnyért is nagy harcot vívnak egymással. Ezért egy vagy két gyalog előny, illetve hátrány legtöbbször már a győzelmet, illetve a vereséget jelenti.

A mozgékonyság szintén ismert fogalom. Az adatok pontos ismertetése és azok összehasonlítása kedvéért háromféle definíció szerint mutatjuk be a függvényeket. Az első meghatározás szerint a mozgékonyság adott helyzetben a lehetséges legális, a második meghatározás szerint a pszeudolegális lépések száma. Ez utóbbiak azokat a lépéseket is tartalmazzák, amelyek sakkban hagyják a királyt. A sakkprogramok általában ezt a megoldást alkalmazzák. Ilyenkor a sakkadást csak a lépés megtétele után ellenőrzik, ami szemben áll a mozgékonysági



3. ábra

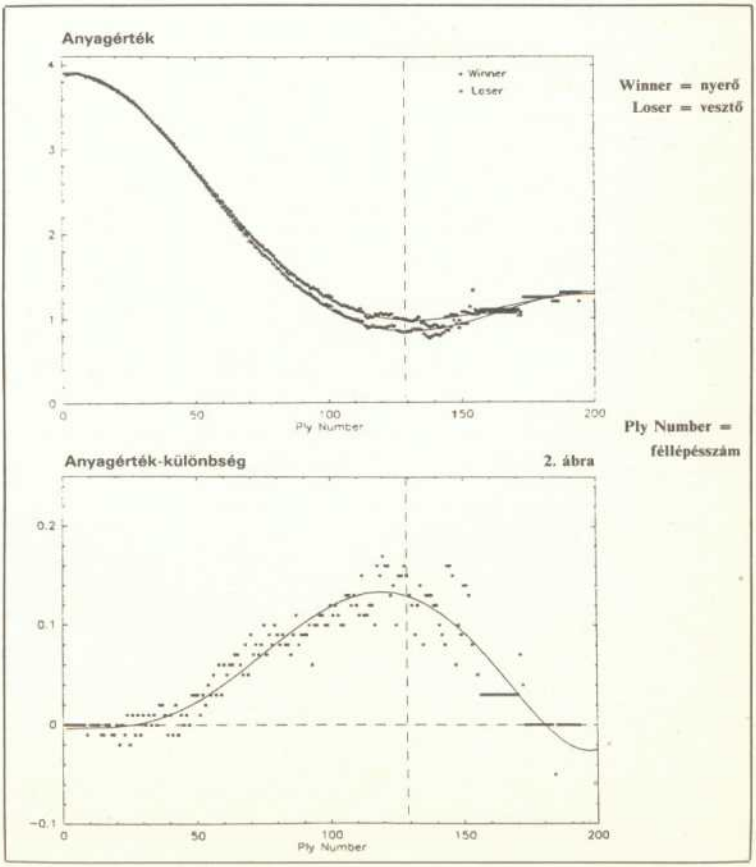
értékkel, hiszen ezt a lépés megtétele előtt számítják ki a lépéslista alapján. A harmadik lehetőségre De Groot professzor mutatott rá, aki a pszeudolegális lépések számát tekinti a mozgékonysági értéknek, de az előző definícióval ellentétben nem értelmezi a függvényt abban az esetben, ha az egyik fél királya sakkban áll.

Nézzük meg a *3. ábrát*, amelyen a Kaszparov—Beljavszki-féle 1981-es játszma 116. féllépése után létrejött hadállást mutatjuk be. Világos és sötét legális lépéseinek a száma egyaránt 31, a pszeudolegális lépések száma világossal 33, sötéttel 36.

A feldolgozott játszmák azt mutatják, hogy a mozgékonysági érték a 40. féllépésben éri el a maximumot, majd csökken a 129. féllépésig. Innen az anyagi érték növekedése miatt a mozgékonysági érték is fokozatosan növekszik. Az e játszmákban előállott pozíciók közül a maximális legális lépések száma 68 volt, a minimálisé 2.

(Folytatjuk)

KOVÁCS P. ATTILA



Az Olvasó írja

A rovatban ezúttal is két levelet olvashatnak, mégpedig azért, mert mindkét levélhez nagyon sok hozzáfűzni valóm van.

Paulo János, Salgótarján

Már régóta halasztgatom, hogy írok önöknek. Azért csak most fogok tollat, mert fellelkesített, hogy egyre többet foglalkoznak az ENTERPRISE számítógéppel. Nagyon szeretnék a gépi kódolásával megismerkedni, de nem tudok a megfelelő szakirodalomhoz hozzájutni. Jó lenne, ha többet írnának róla.

Sokat foglalkozom zenei programokkal, tanulok orgonálni is. Nagyon szeretem Johann Sebastian Bachot. Főleg az ő műveit írom gépre. A szeptemberi számban Dömösi Attila írt egy rövid zenei programot. Számítógépemet azonban, akárcsak a többiét, mint hangszert nem nagyon lehet kihasználni. Visszont mint zenélő számítógéppel fantasztikus hatást lehet elérni. Bach „Nyolc kis preludium és fúga" orgonára írt műveiből már hatot ráírtam a számítógépre. A programok egyenként átlag 28 kilobájtosak. Hármashangzatok szólalnak meg sztereóban. A program, annak ellenére, hogy BASIC-ben van, gyors. Eléri a művekre előírt gyors tempót. A legjobban fejhallgatóval lehet élvezni a zenét.

Ha valakit érdekelnek ezek a programok, szívesen küldök neki. Köszönöm segítségüket.

Az Enterprise-ról még egyszer. Ha olvasónk újra átböngészi a korábbi Mikroszámítógép Magazin-számokat, akkor számos olvasói levelet talál, amelyek dicsérik az Enterprise-ot és a gép programellátottságát, vagy szidják a forgalmazókat, hogy hardvert árulnak és korábbi szoftverigéreteiket nem teljesítik. Sokan engem is elmarasztalnak, hogy az utóbbiak pártján állok, pedig ez valójában nem így van. Mi is a Magazin álláspontja? Nálunk — talán észrevették, ez már hazai szokás — az olvasók érdeke és nem a cégek kívánsága vagy politikája a legfontosabb, hiszen minden kérdésben szeretnénk független és objektív álláspontot kialakítani. Az tény, hogy egyre több Enterprise-számítógép kerül forgalomba, és ezért kötelességünk, hogy az Enterprise-ok amatőr tulajdonosait a lehető legtöbb információval lássuk el. Szerencsére néhány jó tollú professzionális alkalmazó is vásárolt ilyen gépet, így várható, hogy a jövőben még több, az Enterprise otthoni alkalmazásával, illetve szoftver- és harverdlehetőségeivel kapcsolatos írás jelenik meg a Mikroszámítógép Magazinban.

Zene. — Előljáróban: én nem vagyok muzsikus, de az utóbbi időben az NJSZT Számítógépes Zenei Bizottság tagjaival, Patachich Ivánnal és más muzsikusokkal is sokszor dolgoztam együtt. Ezért azt is mondhatom, hogy megszeretettem és így lassan értegetem is a számítógépes zenét. Amiről levelében írt, hogy Bach műveit átírja számítógépre, azt hiszem, ez csak a kezdet, hiszen a számítógépet mint egy új és sokat tudó hangszert használja. Megértem lelkesedését, hiszen mi is átéltük ugyanezt valamikor az ötvenes évek végén, amikor az M—3 először eljátszotta Bach egyik szerzeményét. (Ki tudja, miért — ha jól emlékszem, mi is ezzel kezdtük.) A számítógép ma már nagyon sokféle feladatot lát el, például zenei kutatásban is. Nem is olyan régen alkalmam volt többször is meglátogatni a két leghíresebb párizsi zenei kutató műhelyt, az IRCAM-ot (Institut de Recherche et Coordination Acoustique Musique) és a Xenakis alapította CEMAMU-t (Centre Mathematique, Musique), ahol a zeneszerzés és a zenei kutatás soha nem képzelt lehetőségei állnak a kutatók és a zeneszerzők rendelkezésére.

Visszatérve levelére, az a javaslatom, hogy ne álljon meg Bach műveinek számítógépes interpretálásánál. Ha a számítógép és a zene kapcsolata valóban érdekli, akkor legjobb a hangok mikrovilágába merülnie, amihez igen magas esztétikai érzék is szükséges. Azt hiszem, hogy enélkül ma már nincs esély arra, hogy valaki komoly számítógépet alkalmazó zenei szakember legyen.

Varga Zsigmond, F., Mátészalka

Az első szám megjelenése óta veszem magazinjukat, és kisérem figyelemmel cikkeiket. Nagyon sokat tanultam belőlük. Így jutottam arra az elhatározásra is, hogy gyermekeimnek számítógépet vegyek, ami egy C16-osban öltött testet. Miután élelmiszeripari technológus üzemmérnök vagyok, természetesen érdekelt és érdekel az alkalmazott számítástechnika. Sajnos, szakterületemen olyan nagy a lemaradás, hogy arra szót sem érdemes fordítani, talán az adatfeldolgozásban (adminisztrációban) van mozgás. Ez indított arra, hogy gyermekeim játékra szánt gépét először programokkal „növeltem", majd hardvermódosítással C64-esre változtattam át. Most ott tartok, hogy a SUMMATECH Ipari és Szolgáltató Kisszövetkezet SRD 256 típusú memóriabővítő RAM DISZK-jét próbálom a C64-esről interfésszel a mi gépünkre alkalmassá tenni (az expansions port csatlakozás különbözősége miatt). Természetesen ez már nem gyermekeim lehetőségei szolgálja, hanem az enyémet. Én ugyanis időközben annyira kedvet kaptam a számítástechnikához, hogy elkezdtem tanulni a rendszerszervezői szakon. S bár a kis „játékgép" esetleges tanuláshoz való alkalmazásra már nem jó, de bővítve bizonyos dolgokra még használható. Foglalkoztat az a megoldás is, hogy van már illesztési megoldás az IBM—COMMODORE között, tehát érdemes lenne ezt a lehetőséget megpróbálni, hogy a meglevő expanziós kaput hogyan lehetne felhasználni olyan cartridge csatlakoztatására, amelyikben valamelyik DOS adva van (bár a RAM-DISZK és a lemezmeghajtók is saját operációs rendszerrel dolgoznak).

Az a véleményem, hogy az ország olyan területein, mint ahol én is élek, még sokáig léteznek a C64-es gépek, vannak olyanok is, ahol az adatfeldolgozást most gépesítik, és erre vettek C64-eseket! Ebből kiindulva, sokkal jobban tudatosítani kellene, hogy a korábban említett kisszövetkezet RAM-DISZK-je nagyobb lehetőségeket kínálna azoknak, akik a régi-új C64-esüket szeretnék tovább használni. Talán valamilyen fórumon ezt meg lehetne tenni, valamint azt is, hogy a C64-es IBM-illesztése és hálózatba állítása is megoldott már.

Egy éve, próbaképpen, bevezető jelleggel, a város 5. Sz. Általános Iskolájának 2. osztályájában megpróbáltam számítástechnikát tanítani. Azt hiszem, sikerült felkelteni a gyerekek érdeklődését. Az iskolának külön számítástechnikai kabinetje van. C Plus/4-es és Videoton-számítógépekkel. A pedagógusokkal folytatott beszélgetések során kiderült, hogy keret hiányában semmilyen bővítést nem tudnak csinálni, így pedig alacsony szinten, gyors fejlődés nélkül mehet csak az oktatás. Sokat gondolkoztam azon, hogy a darabszám további növelése helyett egy komolyabb gépnek (egyetlen IBM PC/XT-kompatibilis gépnek) is lennie kellene olyan oktatási intézményekben, ahol komolyan kívánják oktatni a számítástechnikát. Elnézést, hogy így elszaladtam ...

A lapról általában csak jót írhatok. Önöknek köszönhetem, hogy végül is rászántam magam ismét a tanulásra. Köszönöm.

Amikor öt évvel ezelőtt Könyves Tóth Pállal, a Mikromagazint elindítottuk, akkor arra gondoltunk, hogy a lapnak egyik legfontosabb feladata az lesz, hogy a számítástechnikát és a számítógépet megszerettesse a nem számítógépes szakemberekkel is, és annyira közel vigye hozzájuk, hogy ne csak otthon, de a munkahelyükön saját napi feladataik megoldására is felhasználják.

Varga Zsigmond levele nekem — de azt hiszem, a szerkesztőség valamennyi munkatársának — igazi ***ünnepi ajándék.***

Ilyen szép sztorit — amelyet öt éve még elképzelni sem mertünk — csak az élet produkálhat!

Egy élelmiszer-ipari technológus üzemmérnök *olvassa a Mikrót, mert rendszeresen megveszi a gyerekének. A lap cikkeinek hatására ő is vesz — persze csak a gyerekének — egy C16-ot. Ha már a gép megvan, akkor elkezdi használni, majd nem találván kielégítőnek a hardver- és szoftverkörnyezetet, továbbfejleszti a gépet. A levélnek ez a része olyan szakszerű és pontos, hogy csak a vak nem látja: közben az élelmiszeripar szakembere számítástechnikai specialistává fejlődött. — A következő lépés egy fejlettebb számítástechnikai környezet igénye, kirándulás a PC-világba. Ami azt követi, már egyáltalán nem szokványos, ti. át akarja adni tudását azoknak, akiknek leginkább szükségük van rá, a diákoknak. Tanítani kezd és hamarosan világossá válik számára az, amit az ezért felelősök sokszor még mindig nem látnak: az iskolában a mennyiségi szemléletet már régen fel kellett volna váltani a minőséginek. Nem arra kellene törekedni, hogy minél több alkalmatlan számítógép legyen az oktatásban, hanem arra, hogy inkább kevés gép legyen, de az sokat tudjon és el legyen látva szoftverrel és technológiával. Ami persze nemcsak a BASIC tanítására, de a mai számítástechnikában* ***nélkülözhetetlen alkalmazói környezet*** *(táblakezelés, szövegszerkesztés, adatbázisok, hálózat VTX stb.) megismertetésére is alkalmas.*

Végül, még egyszer emlékezve és emlékeztetve a Mikroszámítógép Magazin ***ötödik*** *születésnapjára, ezúton köszöni meg a rovat szerkesztője az Olvasók megtisztelő érdeklődését, azt a sok levelet, amelyet az elmúlt öt évben kapott. Egyben elnézést is kérek azoktól a levélíróktól, akiknek esetleg nem válaszoltunk, illetve leveleiket nem közöltük. Én igyekszem mindig úgy válogatni, hogy a rovatban a közérdeklődésre számottartó leveleket jelentessük meg. A következő öt évben várja leveleiket:*

KOVÁCS GYÖZŐ

Az Ön feje nem adattárolásra való, hanem fontos döntések hozatalára.
Joggal várhatja el, hogy kezében legyen az eszköz, ami munkáját minőségivé teszi, döntését megalapozza.
A számítástechnikában viszont a széles választékból nem könnyű a legjobb mellett dönteni.

az Ön fejével gondolkodtunk,

amikor létrehoztuk az első, Közép-Európában egyedülálló számítástechnikai szalont.
Meghívtuk a legfontosabb gyártókat és forgalmazókat, hogy a választék együtt legyen áttekinthető, kipróbálható, tanulmányozható, összehasonlítható.
Felkészült szakembereink várják az érdeklődőket, a leendő vásárlókat.
Reméljük, döntésünk új korszakot nyit az Ön mindennapi munkájában.

PC szalon

Budapest XIII., Sallai Imre u. 6.
315-136, 310-776
Lépjen új korszakba velünk.
NOVOTRADE

A TII AJÁNLJA

DIÁKOK, FIGYELEM!

Régi s bevált módszer, hogy a diákok a tanév elején beszerzik a könyvek és munkafüzetek mellett a feladatlapokat is, hogy előre felkészülhessenek a dolgozatírásokra.

A taneszközök fejlődése, a számítógépek iskolai alkalmazásának elterjedése ma már új feladatok elé állítja a tanárokat és a diákokat is. Az új tanévben használandó oktatóprogramok már kaphatók a Tudományszervezési és Informatikai Intézetnél.

Intézetünk HT 1080Z, Videoton TV-Computer, Commodore Plus/4, illetve Commodore 64 gépen futtatható programokat kínál kedvezményesen számotokra.

Aki a felsorolt programok közül vásárol, programonként 10%, tíznél több program esetén 20% kedvezményt kap.

Commodore Plus/4-en futtatható oktatóprogramok:

	PROGRAMNÉV	ÁR
MATEMATIKA		
MA/10	Számelmélet	313,0 Ft
MA/11	Egyenlet	313,0 Ft
MA/12	Kombinatorika	313,0 Ft
MA/13	Függvény és transzformáció	313,0 Ft
MA/23	Gyökkereső	313,0 Ft
MA/24	Galton	313,0 Ft
MA/26	Prim	313,0 Ft
MA/28	Rajzolóprogram	375,0 Ft
MA/29	Egyváltozós függvény	375,0 Ft
MA/30	Vektor skalár függvény	375,0 Ft
MA/30	Primitívek	375,0 Ft
MA/48	Négyszögek	313,0 Ft
MA/50	Háromszög koordinátageometriája	313,0 Ft
MA/52	Függvények	313,0 Ft
MA/58	Parabola	313,0 Ft
MA/100	Korrelációk	313,0 Ft
MA/101	Számrendszerek	375,0 Ft
MA/102	Függvény	375,0 Ft
FIZIKA		
FI/03	Soros RC	313,0 Ft
FI/73	Eredő 1	313,0 Ft
FI/74	Eredő 3	313,0 Ft
FI/92	Mágneses mező	313,0 Ft
FI/96	Súlypont	313,0 Ft
KÉMIA		
KÉ/18	Sűrűség	313,0 Ft
KÉ/19	Tömeg	313,0 Ft
KÉ/20	Mólok	313,0 Ft
IDEGEN NYELV		
IN/20	RJAK	313,0 Ft
IN/21	Angol erős igék	313,0 Ft
IN/22	Keyword	313,0 Ft
IN/24	Ürcsata	313,0 Ft
IN/25	Játsszunk a szavakkal	313,0 Ft
FÖLDRAJZ		
FÖ/03—04	SZU—USA	625,0 Ft
TÖRTÉNELEM		
TÖ/13	Magyarország felszabadítása 1—2.	438,0 Ft
MAGYAR IRODALOM		
Mi/02	Vajh, ki ő és merre van hazája	313,0 Ft
MAGYAR NYELVTAN		
MN/35	Helyesírás	375,0 Ft
SZAKKÖZÉPISKOLA		
SK/36	Környezeti tényezők szerepe	313,0 Ft
SK/37	Zöldségfélék csoportosítása	313,0 Ft
SK/38	Zöldségfélék származása	313,0 Ft
SK/39	Összefoglaló-barkochba	313,0 Ft
EGYÉB		
EG/3	Atlétikai teszt	500,0 Ft
EG/04	Ki mit tud?	313,0 Ft
Eg/10	UNIFEL	313,0 Ft
JÁTÉK		
JA/10	Memóriateszt	313,0 Ft

Áraink a 25% áfa-t is tartalmazzák.

Címünk: Tudományszervezési és Informatikai Intézet
1111 Budapest XI., Egry József u. 1—9.
BME „E" épület XI. emelet.
Postai cím: Budapest, Pf.: 454. 1372

A TUDOMÁNYSZERVEZÉSI ÉS INFORMATIKAI INTÉZET előzetes megbeszélés szerint díjmentes programbemutatót tart (vidéken is) az általa forgalmazott oktatóprogramokból. Horváth Zsuzsa 665-011/2663 mellék vagy 813-197 Budapest, Pf. 454, 1372

ÚJ ASSEMBLER A TVC-RE

Ha új, akkor bizonyára régebbi is volt. Most, hogy éppen nem egy oktatóprogramot elemzek — aminek az az oka, hogy a szerkesztőségben a kezembe nyomtak egy programot, nézzem meg —, töprengek, hogyan is fogjak hozzá. Kérdezősködés után kiderült, hogy VIDAS néven már a Videoton is kínálja assemblerét a gépéhez. Kézenfekvő tehát, hogy összehasonlítsuk az újat azzal, ami van.

A program készítője kisvállalkozó, aki termékét a sok mikrogépes vállalkozást igen dinamikusan támogató Magiszter Könyvesbolt útján terjeszti.

A program leírása bizony elég vékonyka, talán azért is, hogy elférjen a kazettatartóban. A VIDAS leírása sem több. Megtaláltam viszont egy olyan Z80 assembler leírását, amit korábban Spectrumon használtam. Ez a SPECTRE—MAC—MON, mely fullscreen editorral integrált mikroassembler. A leírása majdnem harminc oldal. A szövegek körülbelül arányosak a mondanivalóval. A VIDAS leírása csak azért olyan hosszú, mert tartalmaz egy mnemonik táblázatot, egyébként az összehasonlítottak közül a legkevesebb funkció található benne. A programot lényegében a végleges címre helyezi, ami a lehetőségeit korlátozza. Az „új" termék ennél többet tud. A LOAD funkcióval utasíthatjuk, hogy a végleges cím hol legyen.

A VIDAS ténylegesen csak assembler, igaz, annak elég gyors. Az „új" termék viszont, a SPECTRE—MAC—MON-hoz hasonlóan, komplett szoftverfejlesztő rendszer. Ezzel a tulajdonságával hasonlít a Borland cég turbo szoftvereinek filozófiájához, ezért elég korszerű programnak ítélhető. Ha a tárigényeket hasonlítjuk össze, az eltérés szintén szembeötlő. A VIDAS 32 k-s gépen is működik, míg az „új" termék csak 64 k-s gépen tud futni. A szolgáltatásaihoz tehát igen sok tárat elrabol. Ezt nyugodtan felróhatjuk neki, hiszen a több funkciót ismerő SPECTRE—MAC—MON mindössze 16 k-t igényel, pedig nem a Spectrum BASIC editorára hivatkozik, mint akár a VIDAS, akár a TV-Computer Assembler. A nyelvi elemek tekintetében a VIDAS-ban a gépi kód mnemonikjain kívül csak elemi utasításokat használhatunk (ORG, EQU, DB, END). Az „új" termék a már említett LOAD áthelyező utasításon kívül csak néhány újabb adatmezőtípust definiáló utasítást ismer. Ezzel szemben a SPECTRE—MAC—MON ismeri a feltételes fordítás lehetőségét — beleértve a ciklusokat is —, és a szokásos assembler direktivákon túl, mint neve is jelzi, makroprocesszor-lehetőségeket is tartalmaz.

A SPECTRE—MAC—MON-nal akár újradefiniálhatjuk a kódgenerátor-fokozatot is. Amikor használtam, éppen alkalmam nyílt ennek a lehetőségnek a kipróbálására. A programbelövéseknél azt vettem észre, hogy a programomban az egyik ADC utasítás helyett ADD van. Nem így a forráskódban. Nos, meglepődtem egy kicsit, mert hosszú pályafutásom alatt assemblert még nem csíptem rajta hibás működésen. A leírás és a programhoz mellékelt forrásnyelvű makrokönyvtár elemzésével sikerült kiderítenem, hogy az ADC neve nem volt felsorolva az egyik tulajdonságlistán. Ennek kijavításával a programot újrageneráltam, és lássunk csodát, a programhibám megszűnt. A „szerszámom" alkalmas volt arra, hogy még önmagából is kiköszörülhető legyen a csorba.

Mellesleg a forrás- és lefordított program mentő és ellenőrző-betöltő kódrészei egy BASIC modulban voltak. Ez lehetővé tette, hogy a SPECTRE—MAC—MON-t meghajtó helyett át lehessen helyezni 3,5"-os hajlékonylemezre. Ebben a konfigurációban egy Spectrum 128 k-s gépen a meghajtóval elképzelhetetlen fejlesztési sebességet lehetett elérni.

Az „új" termék végül is tartalmaz assembler, disassembler, programlistázási, nyomtatási, monitor (adatbeírási, tárlistázási, tármásolási, tármozgatási), programbelövési (regiszterállapot, töréspontkezelés, lépésenkénti végrehajtás, elindítás) és egyéb funkciókat (munkamező határai, számrendszeralapszám-váltás, kimentés-betöltés a forrás- és tárgykódra).

A használat közben gyorsan visszatérhetünk a BASIC editorba javítás és programbevitel céljából. A program igen jól használható fejlesztő eszköz, még középiskolai iskolagép-környezetben is többet nyújt annál, ami igazán kihasználható. Kevésbé használható profi fejlesztéseknél. Ilyen célra a SPECTRE—MAC—MON-féle megoldás lehet csak igazán jó, aminél szintaktikus vagy címhivatkozási hibánál az assembly leállásakor a kurzor a hibás ponton villog. Ezenkívül legalább a feltételes fordítás az, ami egy profi fejlesztésnél nélkülözhetetlen. Hogy ne is elemezzük a keresztfejlesztést lehetővé tevő makroprocesszor szükségességét. A SPECTRE—MAC—MON talán működne a TVC testvérgépén, az Enterprise-on, Spectrum emulátorral. Ki kellene próbálni, ha még nem tette meg valaki.

Az „új" termék tehát elég korszerű elveken alapuló, jó program, amely iskolagép-környezetben feltétlenül, profi célokra kevésbé, de még így is javasolható.

ZSADÁNYI PÁL

ÖSSZEFOGLALÓ ADATOK

Forgalmazó:	Magiszter Könyvesbolt Bp. V., Városház u. 1.
Terméknév:	TV-Computer Assembler
Szerző:	Kós Géza
Géptípus:	TVC
Hordozó:	kazetta
Dokumentáció:	12 oldalas, a kazettatartóban
Ár:	472 Ft (ÁFA-val)

MINŐSÍTŐ ADATOK

Kezelhetőség:	jó
Teljesség:	jó
Dokumentáltság:	jó
Ár/teljesítmény:	kiváló
Összbenyomás:	jó

KÖNYVEK — HÍREK — ÉRDEKESSÉGEK

Yoneji Masuda:
Az információs társadalom
(Budapest, 1988.
OMIKK,
156 oldal.
Fordította: Hámori Ferenc.)

A japán professzor könyvvé formált látomásában megjelenő „információs társadalom" nem más, mint az ún. posztindusztriális társadalom általa elképzelt változata. Legsajátosabb jellemzője az, hogy tagjai magas szintű intellektuális kreativitás birtokában vannak, hiszen a legmagasabb szinten rendelkezésükre áll minden, a számítástechnika által kínált innovációs lehetőség.

A látomás lenyűgöző! Egészen utópikus fantasztikumnak tűnik ugyan, bár bizonyos részletei a világban már itt-ott kirajzolódni látszanak. Masuda professzor maga is ismertet néhány olyan projektet, amely Japánban, Kanadában, ill. Svédországban a 70-es évek óta működik a közigazgatásban, az oktatásban, a közlekedésben és az egészségügyben. A körvonalakhoz sorolhatók a különböző információs szolgáltató rendszerek is — a kábeltelevíziótól a teletexen át az oktatási és egészségügyi programokig. De vajon mikor jöhet létre teljességében ez a társadalom? A számítások szerint az információs forradalom beteljesüléséhez 64 év elegendő lesz. S 2010 már nincs is olyan messze!

Félő azonban, hogy a további fejlődés nem lesz olyan kiegyensúlyozott, mint amilyennek Masuda feltételezi. Annál is inkább, mert mintha egyáltalán nem számolna — egyebek között — a világ jelenlegi politikai megosztottságával. De azzal sem érthetünk egyet, hogy ő az emberi kiteljesedés kizárólagos színterének a szabadidőt tekinti; a munka világában csak „nyűgöt" lát. Hiszen dolgozni a legutópisztikusabb jövőben is kell majd — hogyan is mondhatna le az ember éppen a munka további humanizálásáról?

Garai Géza:
TV-Computer operációs rendszer
(Budapest, 1988.
Novotrade, 135 oldal.
Ára: 190,— Ft.)

A kötet a Videoton TV-Computer gépi kódú programozásához szükséges ismereteket tartalmazza és segítséget nyújt a BASIC lehetőségeinek jobb kihasználásához is. Foglalkozik a memóriakiosztással, ismerteti a rendszerváltozókat, az operációs rendszer eszközmeghajtóit és azok funkcióit. Az olvasó a programozói munka segítésére hardverismereteket is talál a könyvben.

Külön fejezet dolgozza fel a mikroprocesszorokat és a memóriát, a megszakításkezelést, a be/kiviteli rendszert, a rendszerváltozókat, a kernelhívásokat, a képmegjelenítést, a billentyűzetet, a képernyőszerkesztőt, a nyomtatókezelést és a kazettakezelést.

Purgathofer, Werner:
Grafikai adatok számítógépes feldolgozása.
(Budapest, 1988.
Műszaki Könyvkiadó,
155 oldal. Ára: 98,— Ft.)

A kötet szerzője nem átfogó kézikönyvet kívánt megjelentetni a grafikus adatfeldolgozásról, hanem alapos áttekintést szeretett volna adni a témáról és fontosabb alapfogalmairól. Arra törekedett,

ADOK—VESZEK

Ebben a rovatban rövid, szöveges, a mikroszámítógépekkel kapcsolatos hirdetéseket közlünk. A díjszabás: közületeknek gépelt soronként (60 karakter) 100,- Ft, magánszemélyeknek az első sor 50,- Ft, minden további sor 20,-Ft. Az NJSZT tagjainak az első három sor ingyenes. Hirdetéseiket a szerkesztőség címére várjuk.

ADOK

Atari program idegen nyelvek tanulásának megkönnyítéséhez. Válaszboríték ellenében ingyenes tájékoztató! Dr. Gerő László, Szeged, Budapesti krt. 4/b. VIII/31. 6723

Commodore 128D eladó. Árajánlatokat a következő címre kérek: Donga György, Budapest, Balzac u. 39. 1136

C16, C Plus/4, C64-es programok eladók, 10 Ft/db utánvéttel. Tel.:429-168

C16, Plus/4, C64 programok eladók kazettánként 40-50 program. Egy kazetta 500 Ft. A programok külön is megvehetők 10 Ft/db. Listát előre küldök. Válaszokat válaszborítékkal a következő címre kérek: Szilágyi Imre, Kisvarsány, Rákóczi u. 53. 4811

Commodore 64-hez GYORS-HÁTTÉRTÁR cartridge. Kapacitása 2-31 kbájt. A GYORS-HÁTTÉRTÁR-ba maximum 7 db, célszerűen gyakran használt program vihető be. A gép bekapcsolása után menüvel jelentkezik be, és gombnyomásra a kiválasztott program azonnal fut. Javasolt programcsomagok: Turbo tape, Assembler, Monitor, Supergrafik, Help plus, Turbo másoló. File másoló 1699 Ft és Turbo tape 699 Ft.
Hozott programok elhelyezése a GYORS-HÁTTÉRTÁR-ban. Trompler László, Budapest, Attila u. 22. 1201. Tel.: 287-493 este.

Commodore 64 (új típusú) számítógép magnóval, joystick-kal, német nyelvű kézikönyvekkel és sok minőségi programmal eladó. Ár: 20 000,- Ft. Oláh István, Szolnok, Kassai u. 56. I/4. 5000 Tel.: 56/15-496

C64 + VC 1541 + 1 db joy. + programok lemezen eladók. Árajánlatokat levélben várok. Irimi János, Hajdunánás, Ady E. krt. 33. I/6. 4080

C64-es programkazettáimat és lemezeimet eladnám. Magyar Attila, Kapuvár, Lenin u. 10. 9330

Commodore 64 (új típusú), magnó és 2 db joystick eladó. 653-742 este 6-9-ig.

10 db C64-es programkazettámat 510 db programmal eladom. Egyben 4500,- Ft, esetleg 5-5 db 2500,- Ft-ért. Esetleg videokazettára, illetve floppy lemezekre cserélem. Nagy Márta, Mezőkövesd, Szilhalmi u. 1. 3400

C64, 1541-es floppy, cartridge, 50 diszk a legújabb játék- és felhasználói programokkal, 2 joystick sürgősen, kedvező áron eladó. Beregi, Budapest, II. Törökvész u. 128. vagy 150-149 este 6 után

C64-es új számítógépemet eladom tartozékokkal + TV-Computer (64 k) eladó 200 programmal. Hata József, Tiszaörs, Lenin u. 62. 5262

C64-es programokat adok-cserélek. Új 88-as játékok nagy választékban. Várnai Miklós, Pécs, Egri Gyula u. 84. 7632

C64 (15 000), **VC 1541**(18 000), 2 db holland mikrokapcsolós joystick (4000), 26 lemez játék+szoftver (4000), szakkönyvek (3000), összes Mikro Magazin (1500), együtt 40 000,- Ft-ért eladó. Sárosi Sándor, Környe, Zsíros u. 1/a. 2851, Tel.: 34/72-044 este

Computer Persönlich Magazin 87/88 összes száma eladó: 661-677

Figyelem! Programsegítő modulok, FAST-LOAD, SPEEDTAPE cartridge-k kaphatók C64 számítógéphez. Vállalom továbbá bővített karakterkészlet, Reset gomb, IRQ (Pause) kapcsoló beépítését, Seikosha SP1200VC nyomtató magyarosítását, bioritmus és horoszkóp készítését. Szívesen küldök tájékoztatót! Bártfai Barnabás, Budapest, Móricz Zs. u. 22. 1193

Primo számítógéphez készült új, gépi kódú játékprogramok eladók 170-370 Ft-os áron. A programokat postán, utánvéttel küldöm, esetleg személyesen az esti órákban. Válaszboríték ellenében ingyenes tájékoztatót küldök. Paller Gábor, Budapest, Varga Gy. park 8/a. 1149

Primo A64 tápegységgel, kazettákkal, könyvekkel 6000 Ft-ért eladó. ZX-Spectrum programcsere (keresem a THE LAST NINJA-t). Szántó Péter, Gyömrő, Bercsényi u. 20. 2230

PRINTFOX, CHARAKTERFOX, LUPEFOX és GEOS V1.3 magyar nyelvű dokumentáció eladó. Honti József, Csákvár, Május 1. u. 11. 8083

Teljesen új mikrokapcsolós QuickShot IX jogball eladó. Állítható jobb és balkézre ill. autofire. Irányár: 3000 Ft. Egri Imre, Békésszentandrás, Pacsirta u. 6. 5561

TV-Computer programokat adok, veszek, cserélek. Érdeklődni levélben vagy személyesen vasárnap 9-11 óra között. Csatlós Béla, Mezőtúr, Ifjúsági ltp. XIX. ép. II. lh. 1/3. 5400

Spectrum+ számítógép többszáz programmal, fényceruzával és Enterprise számítógép programokkal sürgősen, olcsón eladó. Kristóf Attila, Teskánd, Petőfi u. 49. 8991

Spectrum (48 k) 10 000 Ft-ért, Commodore 64/128 1541 Mouse 5000 Ft-ért eladó. Magyar Elek, Kecskemét, Dohnányi Ernő u. 54. 6000

Sinclair (128 k) **ZX-Spectrum** + 2 típusú komplett számítógép eladó. Dr. Hajas hétköznap este 895-960, napközben 632-070

Spectrumosok! A Zotyocopy forrása a készítőtől! 200 Ft utánvéttel. Kazetta szükséges. Graff Zoltán, Budapest, Mester u. 61. II/14. 1095

Spectrum (48/128 k) programkazetták a legújabb angol slágerlistákról eladók. Tájékoztatót válaszborítékban küldök. Horváth Péter, Siklós, Pf.: 129. 7800

Spectrumosok figyelem! Eladó néhány 5 1/4-es, 1x 35 track-es MOM floppy drive eredeti gyári csomagolásban, eredeti áron, 3500 Ft-ért. Illesztő építésben is tudunk segíteni! Érdeklődni: Bécsi út - Vörösvári út sarki Virágüzletben (VOLÁN buszvégállomásnál).

ZX-Spectrum (TV hanggal) + 2-es interfész + 300 program eladó. Irányár: 13 500 Ft. Molnár László, Martfű, Lenin u. 18. 5435

ZX-Spectrum szuper (80 k), floppy + Specy Dos interfész, magnó, irodalom együtt vagy külön eladó. Érdeklődni lehet: Varga József, Kerepestarcsa, Szabadság u. 162. 2144

hogy ezt a szerteágazó területet, egyszerűen, jól érthetően mutassa be. A szerző feltételezi az általános adatfeldolgozás alapfogalmainak ismeretét, viszont részletesen tárgyalja a grafikus adatfeldolgozás alapfogalmait. Igyekezett egyszerűen bemutatni a nagyobb összefüggéseket, ami remélhetően segíti egy-egy témakör jobb megértését. A szövegben nem hivatkozott a szakirodalomra, de a függelékben megtalálhatók a kapcsolódó könyvek és folyóiratok címei. A kötet rövid idő alatt két kiadást ért meg.

ZSEBKÖNYVTÁR

Napjainkban egyre gyorsabban terjednek az adattárolás új eszközei, az optikai lemezek. Közülük is leggyorsabban a CD—ROM (Compast Disc — Read Only Memory). Ez voltaképpen a kompakt hanglemeznek a számítógépes adatokat rögzítő „testvére". Sokszorosítása is a kompakt hanglemezhez hasonló: felhasználója tehát írni nem tud rá, hanem csak leolvashatja a gyárilag ráírt információt. Éppen ezért, mert nem változtatható, ideális eszköz arra, hogy kiadóvállalatok számítógéppel olvasható adatbázisokat, kézikönyveket, szótárakat, lexikonokat, programcsomagokat, oktatási csomagokat forgalmazzanak rajta. Tárolási kapacitása óriási: egy-egy 12 centiméter átmérőjű lemezen mintegy félmillió megabájt — azaz mintegy negyedmillió oldal — hasznos információ fér el. A CD—ROM lemezek használatához be kell szerezni egy CD—ROM olvasóberendezést, ami mikroszámítógépekhez csatlakoztatható. Idei amerikai újdonság egy hordozható változat megjelenése: a mikroszámítógép és a CD—ROM-olvasó egy nagyobb aktatáskában is elfér és persze szállítható. Ez a méretcsökkenés már jelentősen kiterjeszti a CD—ROM lemezes kiadványok használhatóságát. Hazánkban is már folynak az előkészületek a CD—ROM széles körű alkalmazására. A CD-lejátszó és egy digitális optikai tárolócsalád kifejlesztéséhez és gyártásának előkészítéséhez a Videoton licencet vásárolt a Thomson cégtől.

CATS

Nem macskákat, hanem számítógéppel támogatott tesztelési rendszert (Computer Aided Testing System — CATS) fejlesztett ki a Logos nyelviskola. A lényege, hogy egy „üres" tesztvázgyűjteményt hoztak létre, melyben az egyes tesztekhez meghatározott számú tesztkérdés tartozik, továbbá előre rögzítették, hogy az egyes kérdésekre hány válasz közül választhat a tanuló, s ezen belül melyik válasz a helyes. Az egyes tesztvázaknak azonosító jelük van, azaz begépelésével hívhatók.

A tesztvázak viszonylag egyszerű munkával gyorsan feltölthetők bármilyen témájú anyaggal. Például az International House Language School angol, német, olasz nyelvismeretet faggató anyaggal töltötte fel a teszteket, és előkészületben vannak a francia, orosz tesztfüzetek is. Ezeket a füzeteket bárki egyszerűen, akár önállóan is használhatja. Természetesen semmi akadálya nincs annak sem, hogy a nyelvoktatástól távoli, akár Kresz-anyaggal töltsük fel a vizsgavázakat.

A program PTA zsebszámológépre, Commodore 64-re és IBM PC-re készült el. Meghökkentő az alacsony ára is: a Commodore 64-es változat tesztenként 600, az IBM PC-s változat pedig 900 forintba kerül.

Két gép az NDK-ból

Megjelentek az első, európai szocialista országból származó IBM PC-vel kompatibilis mikroszámítógépek a hazai piacon. A Migért által kínált A 7150 és ESZ 1834 közös jellemzője, hogy az NDK-beli Robotron kombinát készítette, mikroprocesszoruk az Intel 8086 szovjet megfelelője, 40 Mbájtos winchester tárat tartalmaznak s a hazai árakhoz képest igen olcsók: a nyári induló áruk 180 ezer forint volt.

A Robotron A 7150 típusú mikroszámítógép

Karácsonyi ajánlat

Commodore és Enterprise számítógépek tulajdonosai részére:

Fényceruza C64-hez, C128-hoz 1000 Ft
Fényceruza C16-hoz, Plus/4-hez 1500 Ft
Fényceruza EPROM kártyás kezelőprogrammal C16-hoz és C Plus/4-hez, alkalmazói programokkal, kazettán 2000 Ft
Joystick (C64, C Plus/4, C16 és Enerprise típusokhoz) 625 Ft
Monitor-adapter JUNOSZTY tv-hez (hang és minőségi kép, házilagos szerelhetőség) 400 Ft

Szállítás postai utánvéttel.

Cím: COMPUTEAM GM., Kaposvár, Berzsenyi u. 32. 7400

—CSERÉLEK

ZX-Spectrumra játék- és felhasználói programok eladók. Válaszborítékért katalógust küldök. Bogyó Frigyes, Pécs, Bihari J. u. 3/B. fsz.1. 7633

Személyi számítógéphez illeszthető lemezmeghajtó egység eladó. Nagy Jenő, Győr, Tákó u. 2/A. 9025

VC 1541 eladó. Bölöni Tibor, Várpalota, Hősök tere 2. I/3. 8100

VESZEK

Color Geine, Ge 2000-es számítógéphez gépkönyvet (programozási segédletet) vennék, vagy hosszabb időre kölcsönkérnék. Bokányi József, Zalaegerszeg, Hegyalja u. 58. II/7. 8900

C64 játék- és felhasználói programokhoz keresek leírásokat, programozási ötleteket. Megegyezés levélben. Dukán Zoltán, Sopron, Laktanya u. 20. 9400

1985 előtt gyártott Commodore 64-est veszek. Ormos Zoltán, Budapest 4, 1364 postán maradó

1541 C diszket vennék. Keresem a Microcompiler valamint a Petspeed használati utasítását. Tel.: 202-155 18 óra után

Enterprise gépi kódú programokat vennék. Patek Alajos, Budapest, Faludi u. 28. 1131, Tel.: 291-483

Enterprise programokat veszek, cserélek, keresem a Magic Ball programot. Daróczi Tibor, Tát, Árpád u. 57. 2534

Keresem a Mikroszámítógép Magazin 1983-1986 évfolyamok számait. TURBOPASCAL 4.0 (estleg 3.0) leírást (angol vagy magyar) vásárolnék, vagy másolnék. Csákvári Zoltán, Fűzfő Gyártelep, Gagarin u. 22. 8184

Primóra írt, kazettán leforgatható játékprogramokat veszek. Szele István, Mohács, Jókai u. 1. I/3. 7700

Primóhoz használható sornyomtatót vásárolnék. (DATACOOP, MOM, TMT, FT, KX típusokat). Horváth László 583-544 délelőtt

Sinclair ZX-Spectrum+ (128 k) géphez játék- és felhasználói programokat vennék. Hencz Attila, Győrsövényház, Gárdonyi u. 41. 9161

CSERÉLEK

C16, Plus/4 felhasználói- és játékprogramokat cserélek. Listát kérek! Csókás Csaba, Tamási, Rákóczi u. 47. 7090

C64-es programokat adok-veszek-cserélek kazettán, játék- és felhasználói programok egyaránt érdekelnek. Rácz Balázs, Budapest, Kulpa u. 12. 1025

C64-es programcsere kazettán. Bányai Attila, Tatabánya II. ker. Banyi János u. 3. 2800

C64-es programokat cserélek kazettán. Listát kérek! Hábel Péter, Szolnok, Verseghy u. 38. 5000

C64-es programokat cserélek kazettán. Listát kérek! Bernáth Csaba, Eger, Meder u. 3. 3300

Enterprise programcser. Azok is írhatnak, akiknek kevés programjuk van. Listát kérek! Tóth Zoltán, Cegléd, Sas u. 9. 2700

ZX-Spectrum (48 k) programokat cserélek. Válaszokat kizárólag levélben a következő címre várom: Vágner Gyula, Budapest, Hidegkúti u. 80/a. 1028. Kérem, hogy programlistával válaszoljanak!

ZX-Spectrumra szinvonalas programokat cserélek. Válaszokat listával kérem! Batkó Péter, Jászberény, Sárkány u. 1. 5100

Szinvonalas Spectrum programokat cserélek. Válaszokat listával kérem! Berecz Gábor, Miskolc, Engels u. 38. 3529

Spectrum (48 k) játékprogramokat cserélek. Listát kérek! Dávid Róbert, Bátonyterenye, Bányász u. 13. I/10. 3070

TVC (64 k) tulajdonosokkal leveleznék, programokat cserélnék. Listát kérek! Molnár János, Szolnok, Jászi F. u. 10. VI/25. 5000

TVC-n kb. 450 programmal rendelkezem, hasonló cserepartnereket keresek, listát kérek. Dubrovics Zsolt, Sopron, Frankenburgi u. 48. 9400

Videoton TV-Computerhez játékprogramokat cserélnék! Majer László, Zirc, Akácfa u. 23. 8420

Ez a rovatunk KODEX 2000 szövegszerkesztővel készült.

Pontvadászat

Rejtvénysorozatunk minden fordulójában két feladatot közlünk: az első logikai, matematikai tudást, a második számítástechnikai alapismereteket is igényel.

A feladatok után közöljük az elérhető maximális pontszámot. A részmegoldásokat is pontozzuk.

Ezekkel a feladatokkal a Pontvadászat-sorozatot lezárjuk. A legjobb tíz versenyző nevét magazinunkban közzé is tesszük, ők lesznek azok, akik könyvutalványt is kapnak.

A helyes megoldások a feladatok közreadása után egy lapszámmal később jelennek meg. Ezek után kíváncsian várhatják a nyertesek névsorát.

Beküldési határidő: 1988. december 31.

Címünk: Mikroszámítógép Magazin Szerkesztősége
1371 Budapest, Pf. 433.

Az 1988/6. szám 1. és 2. feladatára 12-12 megfejtés érkezett be. Ezek közül az 1. feladat megfejtési százaléka, a maximális 4 pontot véve 100 százaléknak, 91,7 százalék, a 2. feladaté, a maximális 16 pontot véve 100 százaléknak, 91,1 százalék volt.

Jó vadászatot kíván a feladatok összeállítója:

dr. Hoffmann Tibor

1. feladat

Egy képzeletbeli tengerben van egy A típusú és egy B típusú hal. Az A típusú hal más táplálékon kívül előszerettel fogyasztja a B típusú halat, mégpedig úgy, hogy ahhoz, hogy életben maradhasson, *négy*hetente *5* darab B típusú halat kell felfalnia. Ha életben van, akkor *egy*hetes érési idővel minden második héten minden A hal további *1* hallal szaporodik.

A B típusú halak egyéb kis halakkal táplálkoznak, melyek gyakorlatilag kimeríthetetlen mennyiségben találhatók a tengerben. A B típusú halak érési idő nélkül, hetenként *2* új hallal szaporodnak.

a) *Kérdés: hogyan változik* az A és B típusú *halak aránya*, ha a folyamat kezdetén ugyanannyi A és B típusú hal volt a tengerben? (8 pont)

b) *Hogyan változik a fenti folyamat*, ha a B típusú halak szaporodása csak hetenként *1* új hal? (4 pont)

2. feladat

NO egy adott nagyon nagy egész és X egy 1-nél nagyobb szám. Tekintsük a következő FORTRAN szubrutint:

```
      FUNCTION F(X)
      F=0
      X=(X-1)/(X+1)
      DO 1 I=1, NO
      J=2I-1
      G=Y**J/J
      F=F+G
      IF(G-10**(-9)) 2, 1, 1
1     CONTINUE
2     F=2 F
      RETURN
      END
```

Milyen függvényt ad meg F(X)? (5 pont)

Az 1988/10. szám feladatainak megoldása

1. feladat

Célszerű a számítást úgy végezni, hogy egy alapszázalékhoz sávonként megadjuk, hogy mennyi additív százalékot kell hozzátennünk. Az adósávok alsó határát a_i-vel jelölve, és p_i-vel jelölve a sávban az adó százalékos növekedését, ez a következőképpen írható fel:

$$\left[1-\frac{a_i}{a}+\left|1-\frac{a_i}{a}\right|\right]\frac{p_i}{2},$$

ugyanis a szögletes zárójelen belüli kifejezés *0*, ha $a<a_i$, míg minden más esetben a mellette lévő faktorral szorozva a százalék effektív növekedése.

Látható, hogy a szögletes zárójel növekvő *a*-val alulról közelíti meg a *2*-t, és így a százalék növekedése alulról közelíti meg a p_i-t, de azt soha el nem éri. A most bevezetett értékekkel ezt a százalékos adókulcsot bármely *a*-ra a következő képlettel adhatjuk meg (itt a lehetséges összevonásokat elvégeztük):

$$p=30-\frac{5930000}{a}+10\left|1-\frac{48000}{a}\right|+2{,}5\left(\left|1-\frac{70000}{a}\right|+\right.$$
$$\left.+\left|1-\frac{90000}{a}\right|+\left|1-\frac{120000}{a}\right|+\left|1-\frac{180000}{a}\right|\right)+$$
$$+2\left(\left|1-\frac{150000}{a}\right|+\left|1-\frac{240000}{a}\right|+\left|1-\frac{360000}{a}\right|+\right.$$
$$\left.+\left|1-\frac{600000}{a}\right|+\left|1-\frac{800000}{a}\right|\right).$$

Látható, hogy a progresszivitásban kis inkonzekvencia van a 180 000 és 240 000 Ft közötti évi adóalap esetében (egyenletesen elosztva havi 15 000 és 20 000 Ft között), ahol jobban emelkedik az adókulcs a 150 000 Ft és 180 000 Ft közöttihez (havi 12 500 Ft és 15 000 Ft között) képest, noha ez a következő sávban is ellenkezőleg van. Ennek nem világos az oka. (5 pont)

2. feladat

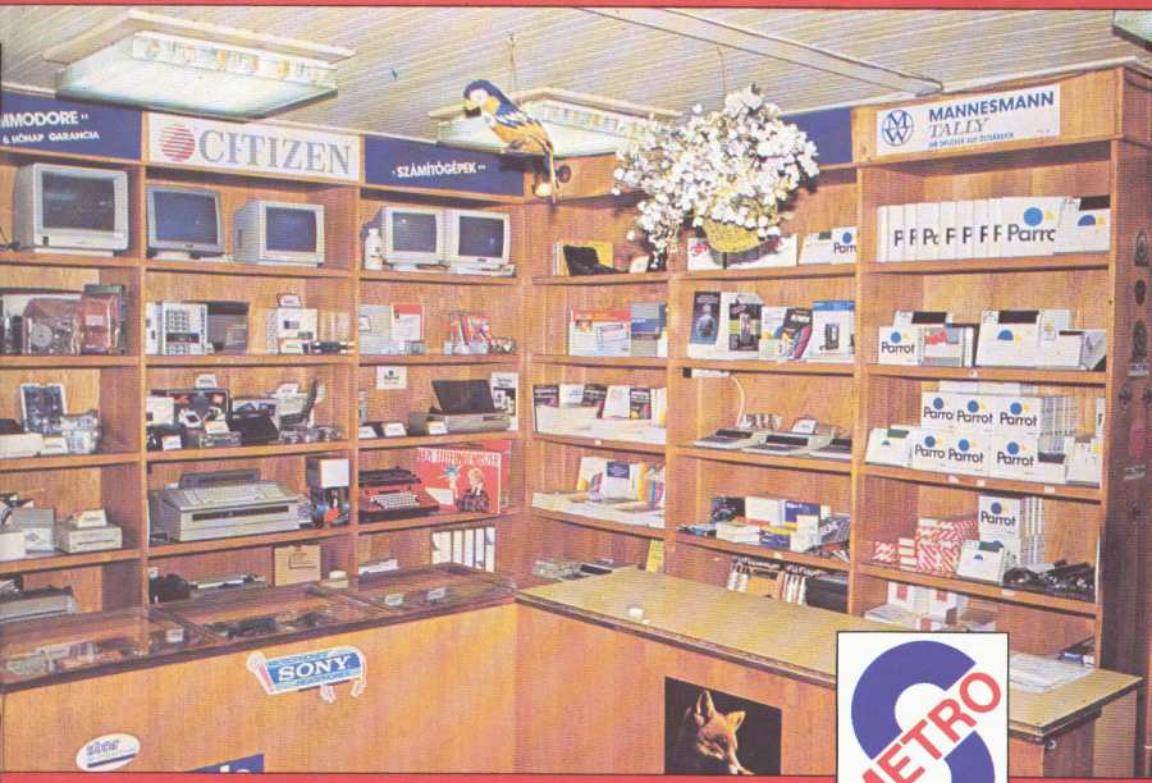
A bitek számának a megduplázásával, vagyis $2(n+1)$ bittel tulajdonképpen kétszer adjuk le ugyanazt a kódot. Ha csak egyetlen hibát veszünk tekintetbe, akkor a feladat megadásában vázolt hibajelző kóddal az első vagy a második kódfélben jelez hibát a rendszer és akkor a második vagy az első kódfél korrekt. (8 pont)

A gyakorlatban vannak ennél kevesebb bit továbbítását igénylő, és ennek megfelelően komplikáltabb korrekciós módszerek is.

A SKÁLA METRÓ ÁRUHÁZ

METRO-COMP üzlete

kiváló minőségű számítástechnikai és híradástechnikai cikkek gazdag választékát kínálja.



Ajánlatunkból:

Parrot floppy lemezek angol importból.
Kiváló minőségben,
mérsékelt áron, széles választékban kaphatók:

5.25" –	DS.DD 48 tpi	180 Ft	3.5" –	DS.DD	400 Ft
–	SS.DD 48 tpi	160 Ft	8" –	SS.DD	220 Ft
–	DS.DD 96 tpi	220 Ft	–	SS.DD	240 Ft
	High Capacity	400 Ft	–	DS.DD	280 Ft

Egyéni és közületi vásárlóinkat egyaránt várjuk a
Budapest VI., Jókai u. 40. szám alatti METRO-COMP üzletünkbe.

Telefon: 127-442 • Telex: 22 7562

VIDEOTON

VT110, VT160, VT180

professzionális személyi számítógépek

A család tagjainak közös jellemzői:

- Az információk max. 4 db háttértárolón rögzíthetők. Ezek lehetnek: floppy diszk, Winchester diszk vagy streamer tárolók.
- A megjelenítés monochrom vagy színes grafikus monitoron történhet.
- Billentyűzetek: 84 nyomógombos PC kompatibilis vagy 113 nyomógombos magyar ékezetes taszt.
- Ajánlott felhasználási területek:
  adatfeldolgozás
  ügyvitel
  szövegszerkesztés és feldolgozás
  műszaki-tudományos számítások
  grafikus alkalmazások
  oktatás
- Lokális hálózati alkalmazások: Ethernet, Arcnet
- Software:
  MS—DOS 3.2 operációs rendszer GWBASIC
  felhasználói programcsomagok nagy választéka